CLUSTERPRO® X SingleServerSafe 3.3
for Linux

設定ガイド

2017.10.02
第6版

CLUSTERPRO

改版履歴

版数	改版日付	内 容
1	2015/02/09	新規作成
2	2015/06/30	内部バージョン 3.3.1-1 に対応
3	2016/01/29	内部バージョン 3.3.2-1 に対応
4	2016/10/03	内部バージョン 3.3.3-1 に対応
5	2017/04/10	内部バージョン 3.3.4-1 に対応
6	2017/10/02	内部バージョン 3.3.5-1 に対応

免責事項

本書の内容は、予告なしに変更されることがあります。
日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任をおいません。
また、お客様が期待される効果を得るために、本書に従った導入、使用および使用効果につきましては、お客様の責任とさせていただきます。
本書に記載されている内容の著作権は、日本電気株式会社に帰属します。本書の内容の一部または全部を日本電気株式会社の許諾なしに複製、改変、および翻訳することは禁止されています。

商標情報

CLUSTERPRO®　は日本電気株式会社の登録商標です。
FastSync®は日本電気株式会社の登録商標です。
Linuxは、Linus Torvalds氏の米国およびその他の国における、登録商標または商標です。
RPMの名称は、Red Hat, Inc.の商標です。
Intel、Pentium、Xeonは、Intel Corporationの登録商標または商標です。
Microsoft、Windowsは、米国Microsoft Corporationの米国およびその他の国における登録商標です。
VERITAS、VERITAS ロゴ、およびその他のすべてのVERITAS 製品名およびスローガンは、VERITAS Software Corporation の商標または登録商標です。
Oracle、JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の 米国およびその他の国における商標または登録商標です。
VMware は、米国およびその他の地域における VMware, Inc. の登録商標または商標です。
Novellは米国および日本におけるNovell, Inc.の登録商標です。
SUSEは米国Novellの傘下であるSUSE LINUX AGの登録商標です。
Citrix、Citrix XenServerおよびCitrix Essentialsは、Citrix Systems, Inc.の米国あるいはその他の国における登録商標または商標です。
WebOTX は日本電気株式会社の登録商標です。
JBossは米国Red Hat, Inc.ならびにその子会社の登録商標です。
Apache Tomcat、Tomcat、Apacheは、Apache Software Foundationの登録商標または商標です。
Androidは、Google, Inc.の商標または登録商標です。
SVFはウイングアークテクノロジーズ株式会社の登録商標です。
本書に記載されたその他の製品名および標語は、各社の商標または登録商標です。

目次

はじめに

対象読者と目的

『CLUSTERPRO® X SingleServerSafe 設定ガイド』は、システムの導入を行うシステムエンジニアと、システム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO X SingleServerSafeの構築作業の手順について説明します。構成は、セクション Ⅰ からセクション Ⅴまでの5部に分かれています。

本書の構成

セクション Ⅰ　CLUSTERPRO X SingleServerSafe の概要

第 1 章「CLUSTERPRO X SingleServerSafe について」：CLUSTERPRO X SingleServerSafe の製品概要について説明します。

セクション Ⅱ　CLUSTERPRO X SingleServerSafeの設定

第 2 章「構成情報を作成する」：WebManager の起動方法、および Builder による構成情報の作成手順をサンプルの構成例を用いて説明します。

第 3 章「システムを確認する」：作成したシステムが正常に動作するかを確認します。

セクション Ⅲ リソース詳細

第 4 章「グループリソースの詳細」：CLUSTERPRO X SingleServerSafe でアプリケーションの制御を行う単位となるグループリソースについての詳細を説明します。

第 5 章「モニタリソースの詳細」：CLUSTERPRO X SingleServereSage で監視を実行する単位であるモニタリソースについての詳細を説明します。

第 6 章「ハートビートリソースの詳細」：ハートビートの詳細について説明します。

第 7 章「その他の設定の詳細」：その他、CLUSTERPRO X SingleServerSafe の設定項目についての詳細を説明します。

セクション Ⅳ 監視のしくみ

第 8 章「監視動作の詳細」：いくつかの障害パターンにより、どのように障害を検出するかの仕組みについての詳細を説明します。

セクション Ⅴ　リリースノート

第 9 章「注意制限事項」：注意事項や既知の問題とその回避策について説明します。

付録

付録 A　「索引」

本書で記述される用語

本書で説明する CLUSTERPRO X SingleServerSafe は、クラスタリングソフトウェアである CLUSTERPRO X との操作性などにおける親和性を高めるために、共通の画面・コマンドを使用しています。そのため、一部、クラスタとしての用語が使用されています。
以下のように用語の意味を解釈して本書を読み進めてください。

用語	説明
クラスタ、クラスタシステム	CLUSTERPRO X SingleServerSafe を導入した単サーバのシステム
クラスタシャットダウン/リブート	CLUSTERPRO X SingleServerSafe を導入したシステムのシャットダウン、リブート
クラスタリソース	CLUSTERPRO X SingleServerSafe で使用されるリソース
クラスタオブジェクト	CLUSTERPRO X SingleServerSafe で使用される各種リソースのオブジェクト
フェイルオーバグループ	CLUSTERPRO X SingleServerSafe で使用されるグループリソース（アプリケーション、サービスなど）をまとめたグループ

CLUSTERPRO X SingleServerSafe マニュアル体系

CLUSTERPRO X SingleServerSafe のマニュアルは、以下の 5 つに分類されます。各ガイドのタイトルと役割を以下に示します。

『CLUSTERPRO X SingleServerSafe インストールガイド』(Install Guide)

CLUSTERPRO X SingleServerSafe を使用したシステムの導入を行うシステムエンジニアを対象読者とし、CLUSTERPRO X SingleServerSafe のインストール作業の手順について説明します。

『CLUSTERPRO X SingleServerSafe 設定ガイド』(Configration Guide)

CLUSTERPRO X SingleServerSafe を使用したシステムの導入を行うシステムエンジニアと、システム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO X SingleServerSafe の構築作業の手順について説明します。

『CLUSTERPRO X SingleServerSafe 操作 ガイド』(Operation Guide)
CLUSTERPRO X SingleServerSafe を使用したシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO X SingleServerSafe の操作方法について説明します。

『CLUSTERPRO X 統合WebManager 管理者ガイド』(Integrated WebManager Administrator's Guide)
CLUSTERPRO を使用したクラスタシステムを CLUSTERPRO 統合WebManager で管理するシステム管理者、および統合WebManager の導入を行うシステム エンジニアを対象読者とし、統合WebManager を使用したクラスタ システム導入時に必須の事項について、実際の手順に則して詳細を説明します。

『CLUSTERPRO X WebManager Mobile 管理者ガイド』(WebManager Mobile Administrator's Guide)
CLUSTERPRO を使用したクラスタシステムを CLUSTERPRO WebManager Mobile で管理するシステム管理者、およびWebManager Mobile の導入を行うシステム エンジニアを対象読者とし、WebManager Mobile を使用したクラスタ システム導入時に必須の事項について、実際の手順に則して詳細を説明します。

本書の表記規則

本書では、注意すべき事項、重要な事項および関連情報を以下のように表記します。

注: は、重要ではあるがデータ損失やシステムおよび機器の損傷には関連しない情報を表します。

重要: は、データ損失やシステムおよび機器の損傷を回避するために必要な情報を表します。

関連情報: は、参照先の情報の場所を表します。

また、本書では以下の表記法を使用します。

表記	使用方法	例
[ ] 角かっこ	コマンド名の前後 画面に表示される語 (ダイアログ ボックス、メニューなど) の前後	[スタート] をクリックします。 [プロパティ] ダイアログ ボックス
コマンドライン中の [ ] 角かっこ	かっこ内の値の指定が省略可能であることを示します。	`clpstat -s[-h host_name]`
#	Linux ユーザが、root でログインしていることを示すプロンプト	`# clpcl -s -a`
モノスペースフォント (courier)	パス名、コマンド ライン、システムからの出力 (メッセージ、プロンプトなど)、ディレクトリ、ファイル名、関数、パラメータ	`/Linux/3.3/jp/server/`
モノスペースフォント**太字** (courier)	ユーザが実際にコマンドプロンプトから入力する値を示します。	以下を入力します。 **`clpcl -s -a`**
モノスペースフォント斜体 (courier)	ユーザが有効な値に置き換えて入力する項目	`rpm -i clusterprosssbuilder-<`*バージョン番号*`>-<`*リリース番号*`>.i686.rpm`

最新情報の入手先

最新の製品情報については、以下のWebサイトを参照してください。

http://jpn.nec.com/clusterpro/

セクション I　CLUSTERPRO X SingleServerSafe の概要

このセクションでは、CLUSTERPRO X SingleServerSafe の製品概要と監視機能概要について説明します。

第 1 章　CLUSTERPRO X SingleServerSafe について

第 1 章　CLUSTERPRO X SingleServerSafe について

本章では、CLUSTERPRO X SingleServerSafe の機能概要の説明と、監視可能な障害について説明します。

本章で説明する項目は以下のとおりです。

CLUSTERPRO X SingleServerSafe とは?

CLUSTERPRO X SingleServerSafe は、サーバにセットアップすることで、サーバ上のアプリケーションやハードウェアの障害を検出し、障害発生時には、アプリケーションの再起動やサーバの再起動を自動的に実行することで、サーバの可用性を向上させる製品です。

通常のサーバでは、アプリケーションが異常終了した場合、アプリケーションの終了に気づいた時点で、アプリケーションの起動を手動で行う必要があります。

また、アプリケーションは異常終了していないが、アプリケーション内部での動作が不安定になり正常に動作していない場合があります。このような異常状態になっていることは、通常では容易に知ることはできません。

ハードウェア障害が発生した場合、一時的な障害であれば、サーバの再起動で正常に戻る可能性があります。しかし、ハードウェア障害に気づくのは困難で、アプリケーションの動作がどうもおかしいと調査を行った結果、ハードウェア障害であったということがよくあります。

CLUSTERPRO X SingleServerSafe では、異常を検出したいアプリケーション、ハードウェアを指定することで、自動的に障害を検出し、自動的にアプリケーション、サーバの再起動を行うことで、障害からの復旧処理を行います。

注: 上述のようにハードウェアの物理的な障害に関しては、サーバの再起動では復旧できないことが多いです。ハードウェアの物理的障害に備えるには、ハードウェアの二重化やクラスタリングソフトなどの導入を検討してください。

CLUSTERPRO X SingleServerSafe の障害監視のしくみ

CLUSTERPRO X SingleServerSafe では、各種監視を行うことで、迅速かつ確実な障害検出を実現しています。以下にその監視の詳細を示します。

- アプリケーションの死活監視

  アプリケーションを起動用のリソース (アプリケーションリソース、サービスリソースと呼びます) により起動し、監視用のリソース (アプリケーション監視リソース、サービス監視リソースと呼びます) により定期的にプロセスの生存を確認することで実現します。業務停止要因が業務アプリケーションの異常終了である場合に有効です。

  注 1: CLUSTERPRO X SingleServerSafe が直接起動したアプリケーションが監視対象の常駐プロセスを起動し終了してしまうようなアプリケーションでは、常駐プロセスの異常を検出することはできません。

  注 2: アプリケーションの内部状態の異常 (アプリケーションのストールや結果異常) を検出することはできません。

- 監視オプションによるアプリケーション/プロトコルのストール/結果異常監視

  別途ライセンスの購入が必要となりますが、データベースアプリケーション(Oracle,DB2 等)、プロトコル(FTP,HTTP 等)、アプリケーションサーバ(Websphere,Weblogic 等)のストール/結果異常監視を行うことができます。詳細は、「第 5 章 モニタリソースの詳細」を参照してください。

- リソースの監視

  CLUSTERPRO X SingleServerSafe のモニタリソースにより各種リソース(アプリケーション、サービスなど)や LAN の状態を監視することで実現します。業務停止要因が業務に必要なリソースの異常である場合に有効です。

監視できる障害と監視できない障害

CLUSTERPRO X SingleServerSafe には、監視できる障害とできない障害があります。クラスタシステム構築時、運用時に、どのような監視が検出可能なのか、または検出できないのかを把握しておくことが重要です。

業務監視で検出できる障害とできない障害

監視条件: 障害アプリケーションの消滅、 継続的なリソース異常、 あるネットワーク装置への通信路切断

- 監視できる障害の例
  - アプリケーションの異常終了
  - LAN NIC の故障
- 監視できない障害の例
  - アプリケーションのストール/結果異常

    アプリケーションのストール/結果異常を CLUSTERPRO X SingleServerSafe で直接監視することはできませんが、アプリケーションを監視し異常検出時に自分自身を終了するプログラムを作成し、そのプログラムを EXEC リソースで起動、PID モニタリソースで監視することで、再起動を発生させることは可能です。

セクション II　CLUSTERPRO X SingleServerSafe の設定

このセクションでは、CLUSTERPRO X SingleServerSafe の設定手順を示します。設定構成例として、一般的な構成となるアプリケーション制御と IP 監視の設定を行います。

第 2 章　　構成情報を作成する
第 3 章　　システムを確認する

第 2 章　　構成情報を作成する

CLUTERPRO X SingleServerSafe では、構成内容を記述するデータのことを、構成情報と呼びます。通常は、WebManager から起動した Builder を用いて構成情報を作成します。本章では、WebManager の起動方法、および Builder による構成情報の作成手順をサンプルの構成例を用いて説明します。

本章で説明する項目は以下のとおりです。

設定値を確認する

Builder(WebManager の設定モード)を使用して実際に構成情報を作成する前に、構成情報として設定する値を確認します。値を書き出して、情報に漏れがないかを確認しておきましょう。

環境のサンプル

以下に、構成情報のサンプル値を記載します。以降のトピックでは、この条件で構成情報を作成する手順をステップ バイ ステップで説明します。実際に値を設定する際には、構築する構成情報と置き換えて入力してください。値の決定方法については、「第 4 章 グループリソースの詳細」「第 5 章 モニタリソースの詳細」を参照してください。

構成設定例

設定対象	設定パラメータ	設定値
サーバの情報	サーバ名	server1
	モニタ リソース数	3
グループ	タイプ	フェイルオーバ
	グループ名	failover1
	起動サーバ	server1
1 つ目のグループリソース	タイプ	EXECリソース
	グループ リソース名	exec1
	常駐タイプ	常駐
	開始パス	実行ファイルのパス
1 つ目のモニタリソース (デフォルト作成)	タイプ	ユーザ空間監視
	モニタ リソース名	userw1
2 つ目のモニタリソース	タイプ	IP監視
	モニタリソース名	ipw1
	監視 IP アドレス	192.168.0.254(ゲートウェイ)
	回復対象	LocalServer
	再活性しきい値	-
	最終動作	サービス停止とOS再起動
3 つ目のモニタリソース	タイプ	PID監視
	モニタリソース名	pidw1
	対象リソース	exec1
	回復対象	failover1
	再活性しきい値	3
	最終動作	サービス停止とOS再起動

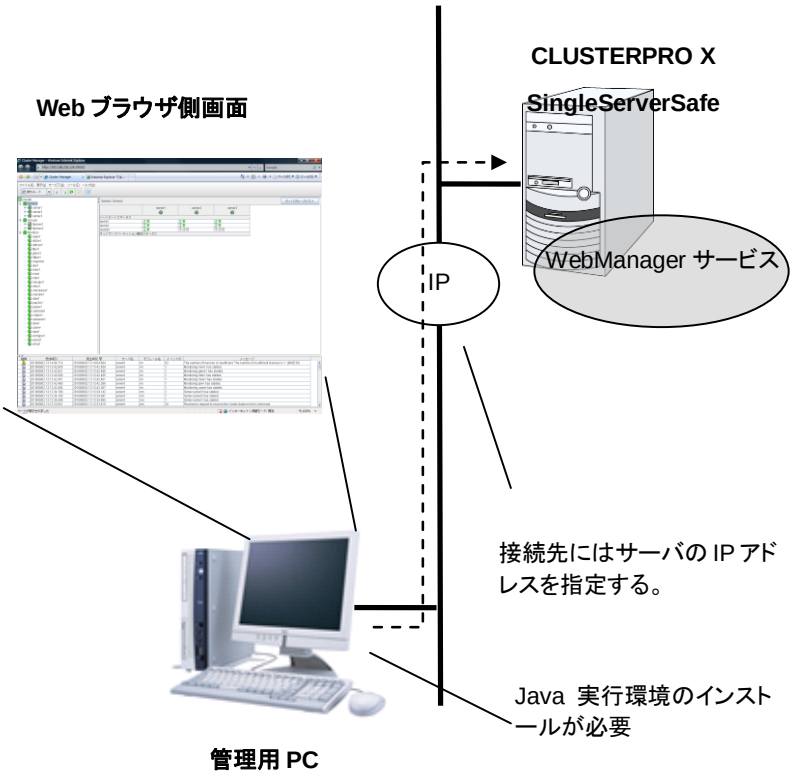
注: 1 つ目のモニタリソースの「ユーザ空間監視」は自動的に設定されます。

WebManager を起動する

構成情報を作成するには、WebManager にアクセスする必要があります。ここでは、まず WebManager の概要を説明し、その後、WebManager にアクセスして、構成情報を作成する方法について説明します。

WebManager とは

WebManager とは、Web ブラウザ経由で Builder(WebManager の設定モード)への切り替え、サーバの状態監視、サーバ/グループの起動/停止及び、動作ログの収集などを行うための機能です。以下の図に WebManager の概要を示します。



CLUSTERPRO X SingleServerSafe のサーバ上の WebManager サービスは OS の起動と同時に起動するようになっています。

管理用PCへの Java 実行環境の設定

WebManager に接続するためには、管理用 PC の Web ブラウザに Java プラグイン（Java™ Runtime Environment Version 6.0 Update 21(1.6.0_21) 以降、Java™ Runtime Environment Version 7.0 Update 2(1.7.0_2)以降）、または Java™ Runtime Environment Version 8.0 Update 5(1.8.0_5)以降） がインストールされている必要があります。

管理用 PC にインストールされている Java プラグインのバージョンが上記よりも古い場合、ブラウザから Java のインストールを促されることがあります。この場合、CLUSTERPRO の WebManager で動作確認されているバージョンの Java プラグインをインストールしてください。

Web ブラウザに Java プラグインを組み込む方法については、Web ブラウザのヘルプ、並びに JavaVM のインストールガイドを参照してください。

WebManager を起動するには

WebManager を起動する手順を示します。

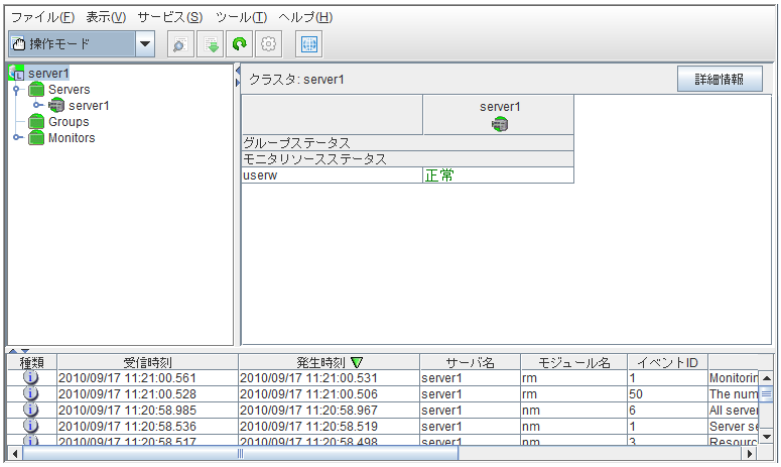
- Web ブラウザを起動します。
  ブラウザのアドレス バーに、CLUSTERPRO X SingleServerSafe をインストールしたサーバの IP アドレスとポート番号を入力します。

  http://192.168.0.3:29003/

  インストール時に指定したWebManager のポート番号を指定します(既定値29003)。

  CLUSTERPRO X SingleServerSafeをインストールしたサーバのIPアドレスを指定します。
  自サーバの場合は、localhostでも問題ありません。

- WebManager が起動します。



- [表示] メニューから [設定モード] をクリックして、設定モード（オンライン版 Builder）に切り替えます。

構成情報の作成手順

構成情報を作成するには、サーバの設定、グループの作成、モニタリソースの作成の 3 つのステップを踏みます。新規に構成情報を作成する場合は、生成ウィザードを使います。以下に手順の流れを示します。

注: 作成した構成情報のほとんどは名称変更機能やプロパティ表示機能を使用して後から変更できます。

1　サーバの設定

CLUSTERPRO X SingleServerSafe を動作させるサーバを設定します。

1-1　サーバを設定する

構築するサーバ名などを設定します。

2　グループの設定

グループを作成します。グループでアプリケーションの起動・終了を制御します。必要な数のグループを作成します。通常、制御したいアプリケーション数ほど必要ですが、「スクリプトリソース」を使用した場合は、1つのグループで複数のアプリケーションをまとめることもできます。

2-1　グループを追加する

グループを追加します。

2-2　グループ リソースを追加する

アプリケーションの起動・終了を行うリソースを追加します。

3　モニタリソースの設定

指定された監視対象を監視する、モニタリソースを追加します。
監視したい数、作成します。

3-1　モニタリソースを追加する

監視を行うモニタリソースを追加します。

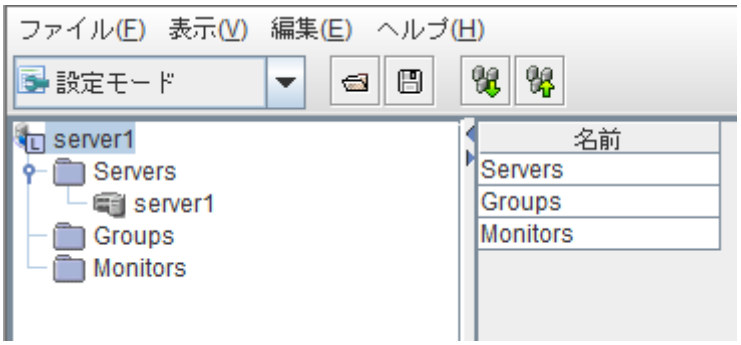
1. サーバの設定

サーバを設定します。

1-1 サーバを設定する

CLUSTERPRO X SingleServerSafe をインストール後、OS を再起動することで自動的に作成されます。WebManager の操作モードから設定モード（オンライン版 Builder）画面に切り替えると既に作成済みの情報が表示されます。

テーブルビューは以下のようになっています。



2. グループの設定

グループとは、システム内のある 1 つの独立した業務を実行するために必要なサービスやプロセスの集まりのことです。

グループを追加する手順を説明します。

2-1 グループを追加する

グループの設定を行います。

1. ツリー ビューの [Groups] をクリックし、[編集] メニューの [追加] をクリックします。

2. [グループの定義] 画面が開きます。

以下のタイプから、選択してください。

タイプ

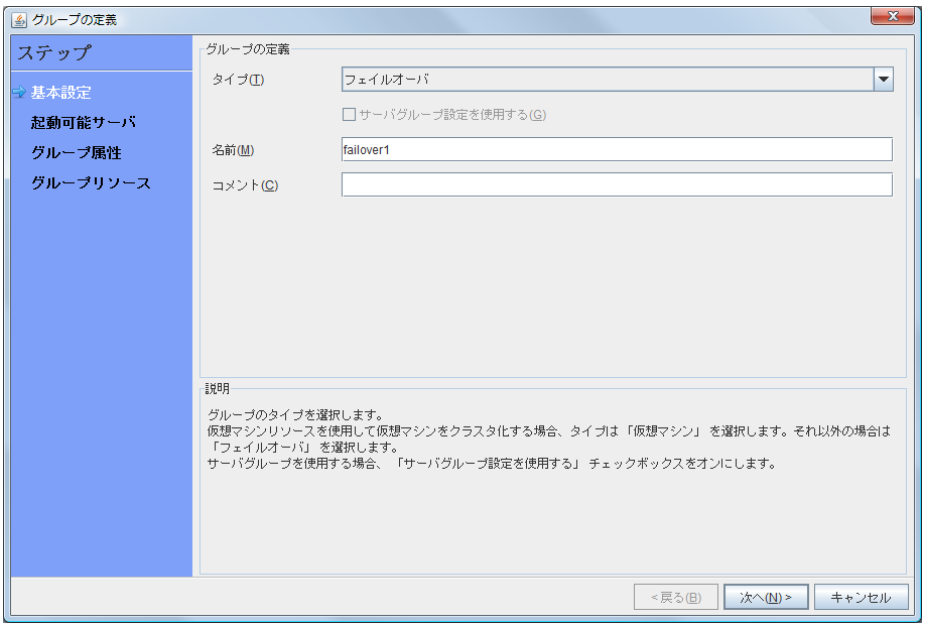
- フェイルオーバ

  通常はこちらのタイプを選択します。

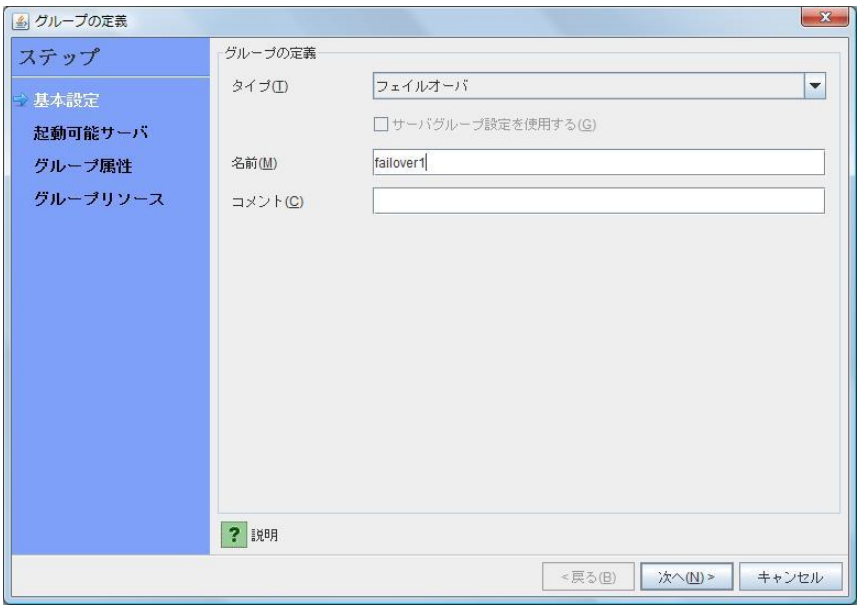
- 仮想マシン

  仮想マシンリソースを使用する場合はこちらのタイプを選択します。

3. [名前] ボックスにグループ名 (failover1) を入力し、[次へ] をクリックします。

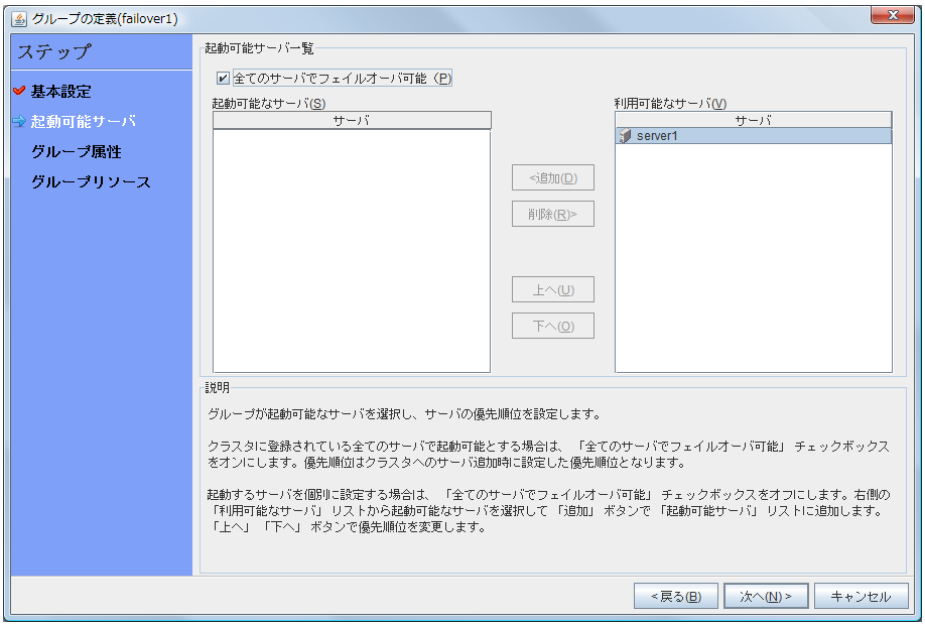


解像度が 800×600 以下の場合、説明欄はツールチップとして表示されます。

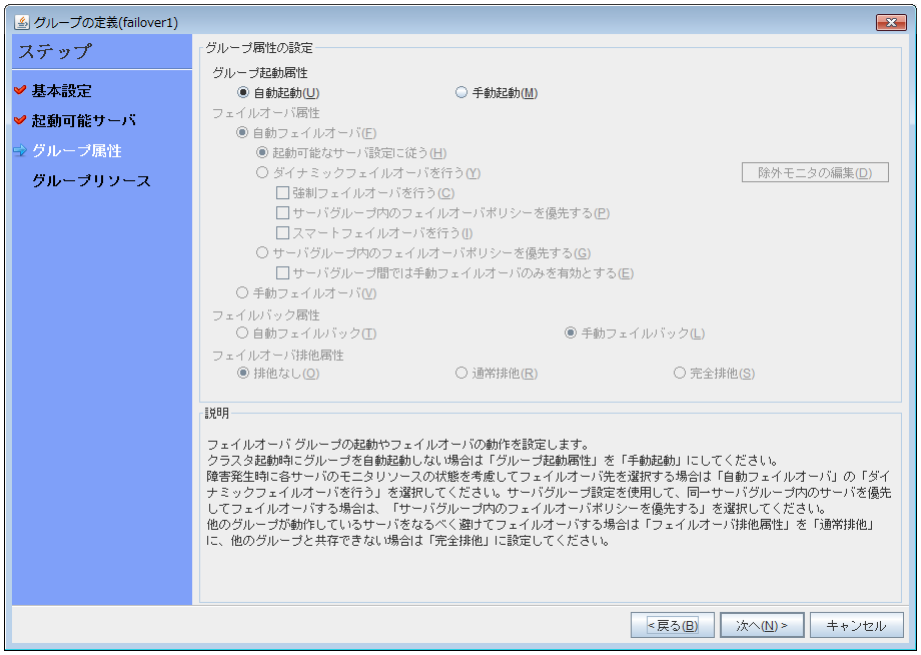


[?] アイコン上にマウスを移動すれば、ツールチップで説明が表示されます。

4. [全てのサーバでフェイルオーバ可能] チェックボックスのチェックがオンになっていることを確認し、[次へ] をクリックします。



5. グループの各属性値を設定する画面です。そのまま [次へ] をクリックします。



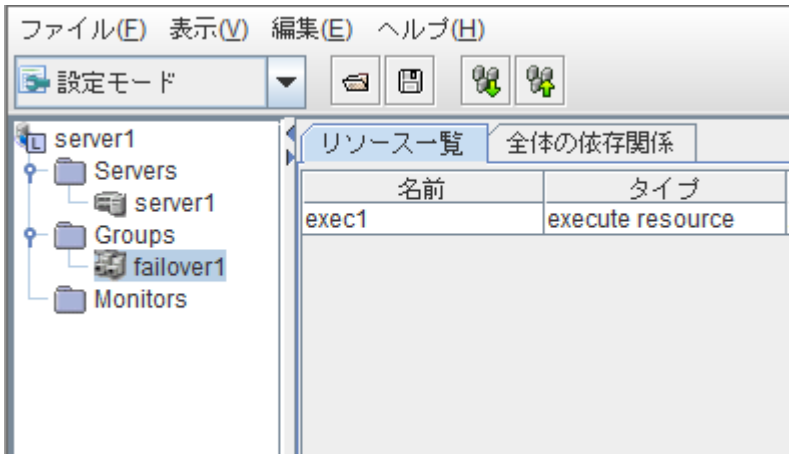
6. [グループリソースの定義一覧]が表示されます。そのまま [完了] をクリックします。
テーブルビューは以下のようになります。

2-2 グループ リソース (EXEC リソース) を追加する

スクリプトによってアプリケーションの起動/終了を行う、EXEC リソースを追加します。

1. ツリー ビューの [failover1] をクリックし、[編集] メニューの [追加] をクリックします。
2. [リソースの定義] ダイアログ ボックスが開きます。[タイプ] ボックスでグループ リソースのタイプ (execute resource) を選択し、[名前] ボックスにグループ名 (exec1) を入力します。[次へ] をクリックします。
3. 依存関係設定が表示されます。何も指定せず [次へ] をクリックします。
4. 復旧動作設定が表示されます。[次へ] をクリックします。
5. [ユーザアプリケーション] をチェックします。また、[Start path]に、実行ファイルのパスを指定します。
6. [調整]をクリックし、ダイアログボックスを開きます。[開始スクリプト]で、[非同期]をチェックし、[OK]をクリックします。
7. [完了] をクリックします。

   テーブルビューは以下のようになります。

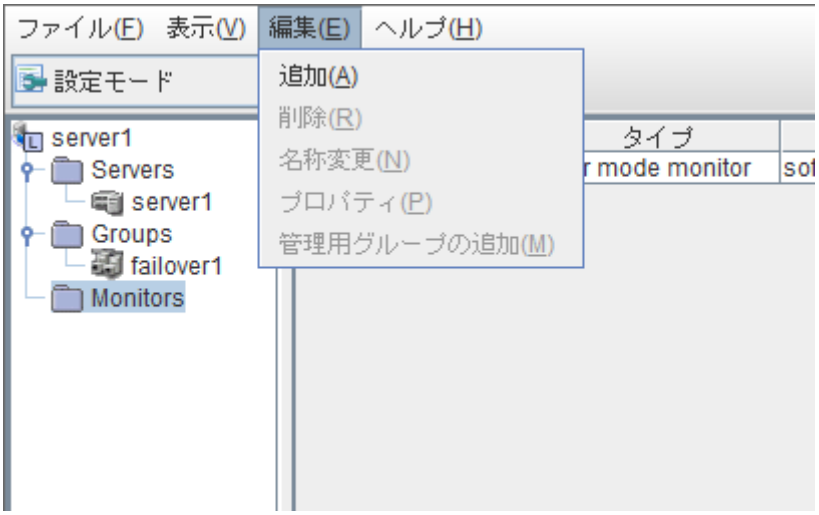


3. モニタリソースの設定

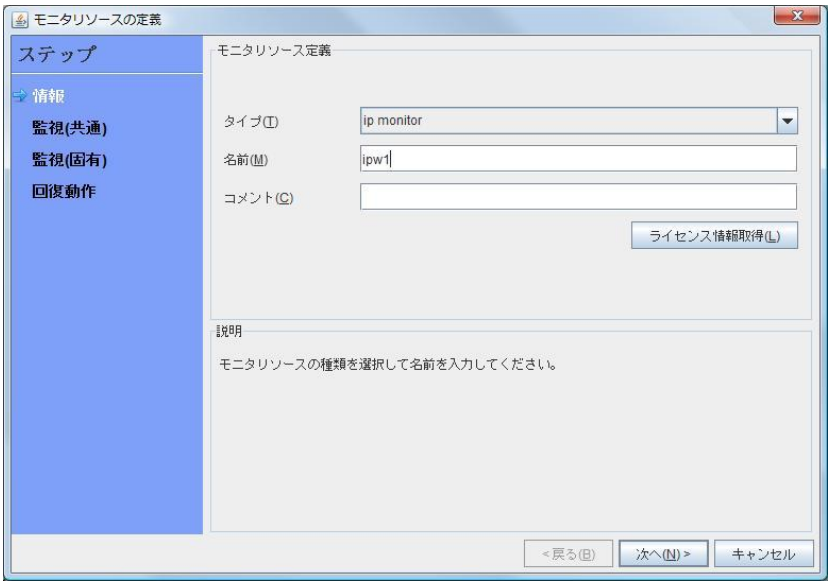
指定した対象を監視するモニタリソースを追加します。

3-1 モニタリソース (IP 監視リソース) を追加する

1. ツリービューの Monitors オブジェクトをクリックし、[編集] メニューの [追加] をクリックします。[モニタリソースの定義一覧] が表示されます。

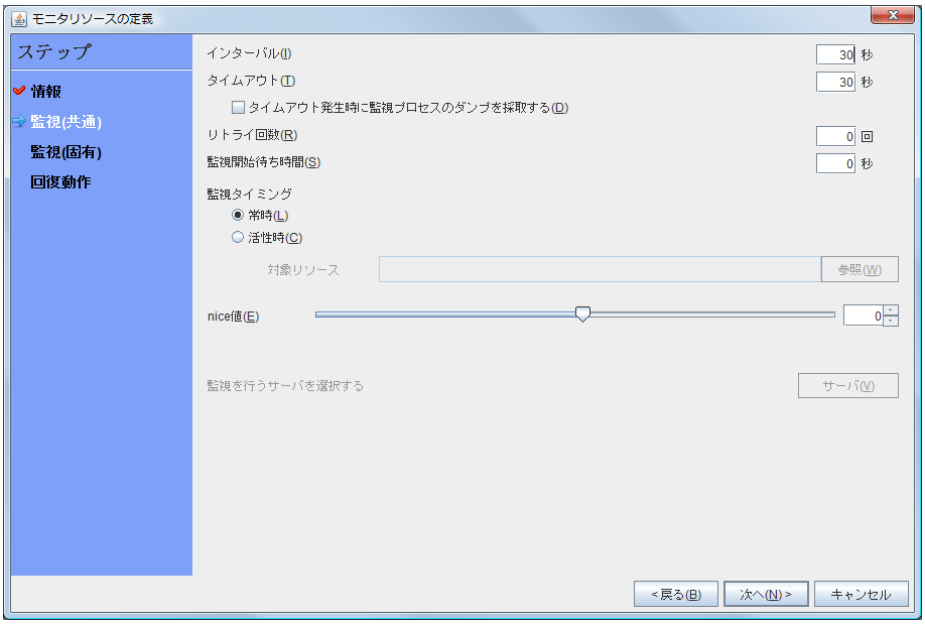


2. [タイプ] ボックスでモニタリソースのタイプ (IP 監視) を選択し、[名前] ボックスにモニタリソース名 (ipw1) を入力します。[次へ] をクリックします。

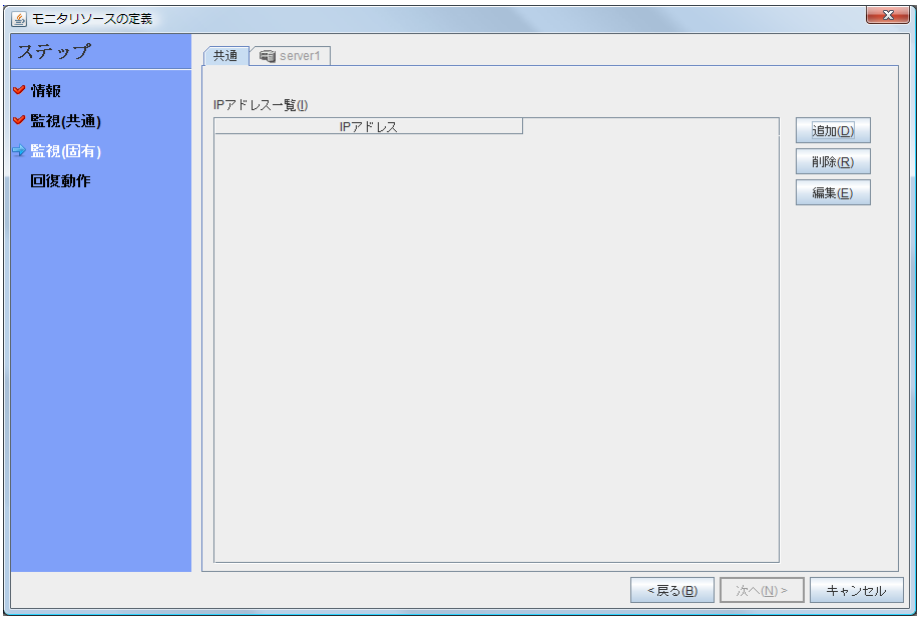


注: タイプとして、モニタリソースが表示されるので、監視したいリソースを選択します。
オプション製品のライセンスがインストールされていない場合、ライセンスに対応するリソースおよび監視リソースは Builder（オンライン版）の一覧に表示されません。
インストールされているライセンスが表示されない場合、[ライセンス情報取得] をクリックしてライセンス情報を取得してください。

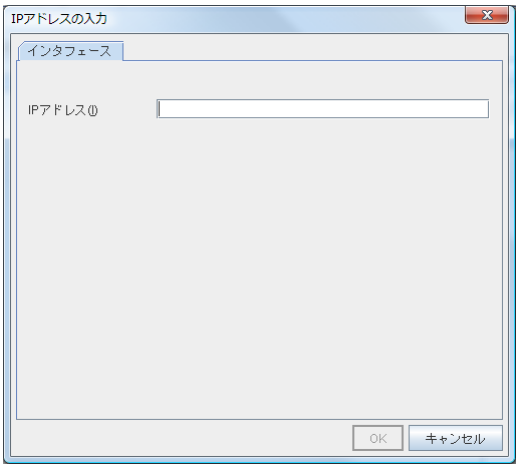
3. 監視(共通)設定を入力します。ここではデフォルト値のまま変更せず、[次へ] をクリックします。



4. [IP アドレス一覧] が表示されます。[追加] をクリックします。

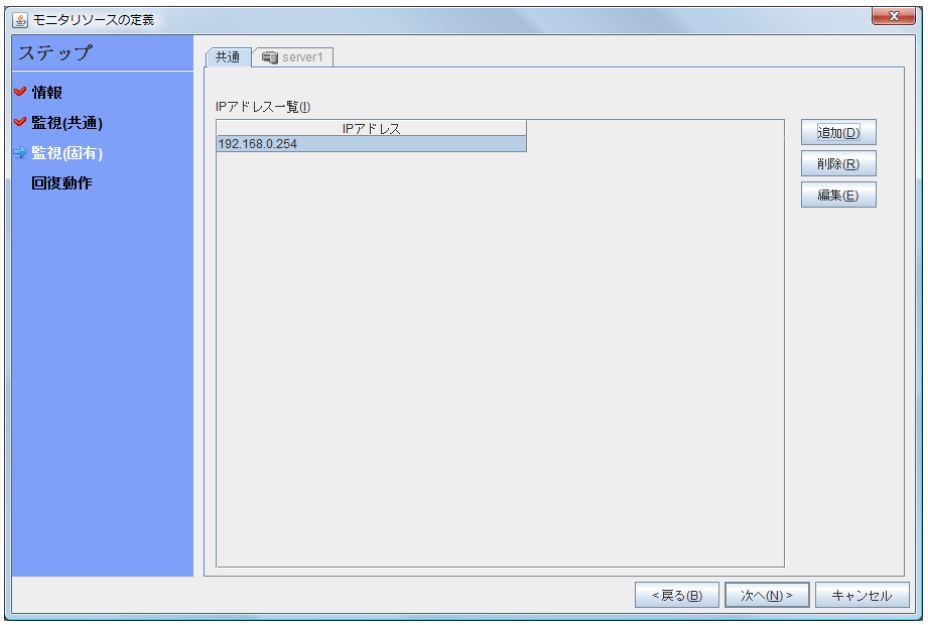


5. [IP アドレス] ボックスに監視 IP アドレス (192.168.0.254) を入力し [OK] をクリックします。

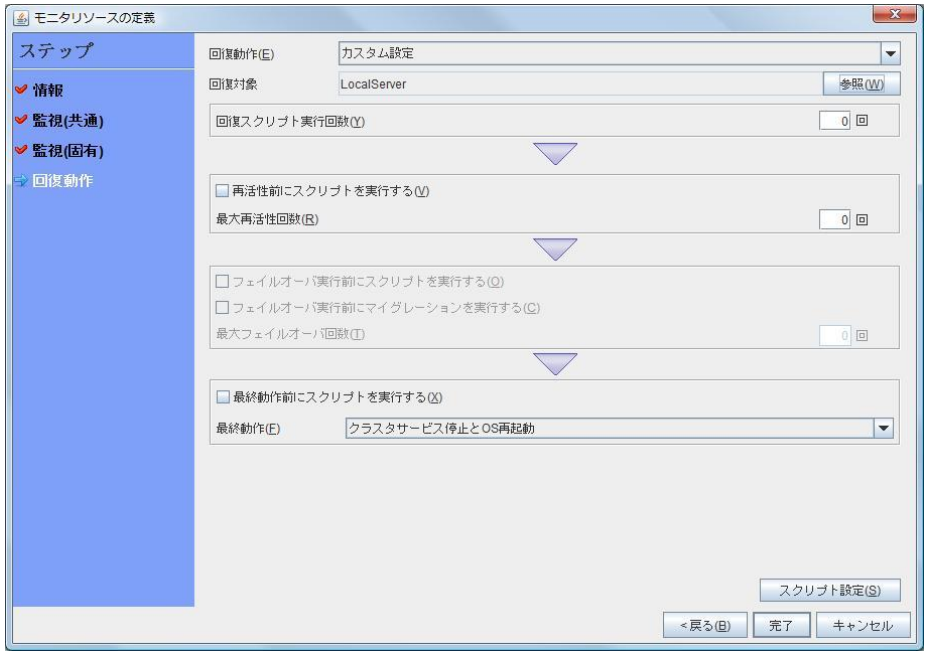


注: IP 監視リソースの監視対象には、LAN 上で、常時稼動が前提とされている機器 (例えば、ゲートウェイ) の IP アドレスを指定します。

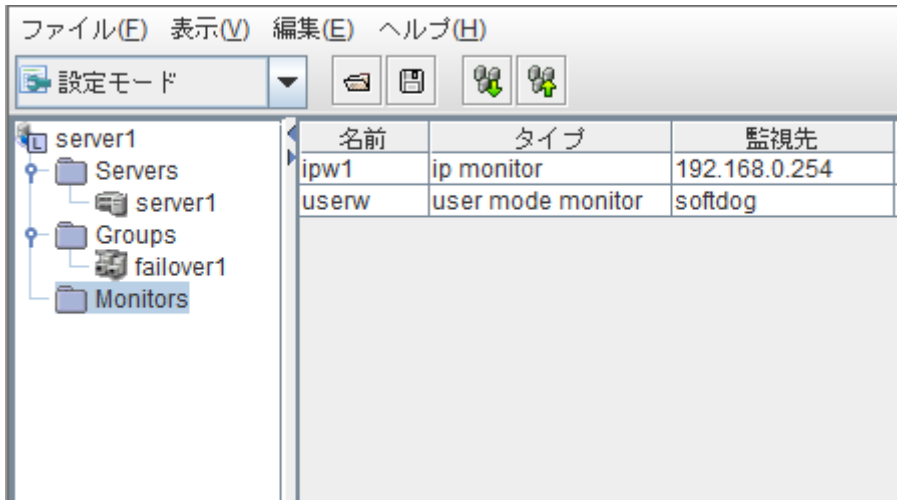
6. 入力した IP アドレスが [IP アドレス一覧] に設定されます。[次へ] をクリックします。



7. 回復動作設定が表示されます。[参照]を押して LocalServer を選択します。[完了] をクリックします。



設定後の画面は以下のようになります。



3-2 モニタリソース (PIDモニタリソース) を追加する

1. このモニタリソースは EXEC リソースの開始スクリプトの種類が[非同期]の場合に設定可能です。
2. ツリービューの Monitors オブジェクトをクリックし、[編集]→[追加] を選択します。[タイプ] ボックスでモニタリソースのタイプ (pid monitor) を選択し、[名前] ボックスにモニタリソース名 (pidw1) を入力します。[次へ] をクリックします。
3. 監視(共通)設定を入力します。[参照] をクリックします。
4. 表示されるツリー ビューで [exec1] をクリックし、[OK] をクリックします。[対象リソース] に [exec1] が設定されます。[次へ] をクリックします。
5. 回復対象を設定します。[参照] をクリックします。
6. 表示されるツリー ビューで [failover1] をクリックし、[OK] をクリックします。[回復対象] に [failover1] が設定されます。
7. [完了] をクリックします。

   テーブルビューは以下のようになります。

以上で構成情報の作成は終了です。次の「構成情報を保存する」へ進んでください。

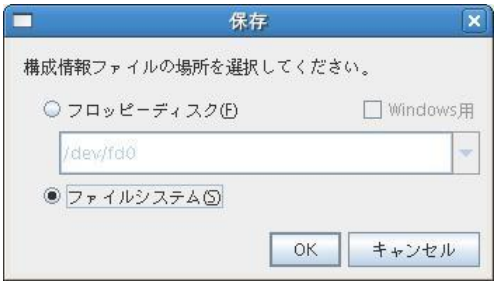
構成情報を保存する

構成情報は、ファイル システム上に保存する方法と、FD 等のメディアに保存する方法があります。WebManager 経由で Builder を起動している場合、保存した情報を CLUSTERPRO Server をインストールしたサーバマシンに WebManager 経由で反映させることができます。

構成情報をファイル システムへ保存する (Linux)

Linux マシン使用時に、ファイル システムに構成情報を保存するには、以下の手順に従ってください。

1. Builder の [ファイル] メニューから [設定のエクスポート] をクリックするか、 ツールバーの [] をクリックします。
2. 以下のダイアログ ボックスで [ファイルシステム] をクリックし、[OK] をクリックします。



3. 以下のダイアログ ボックスが表示されます。任意のディレクトリを選択または作成し、[保存] をクリックします。

注 1: 保存されるのはファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) です。これらのファイルとディレクトリがすべて揃っていない場合はクラスタ生成コマンドの実行が不成功に終わりますので、移動する場合はかならずこの 2 点をセットとして取り扱ってください。なお、新規作成した構成情報を変更した場合は、上記 2 点に加えて clp.conf.bak が作成されます。

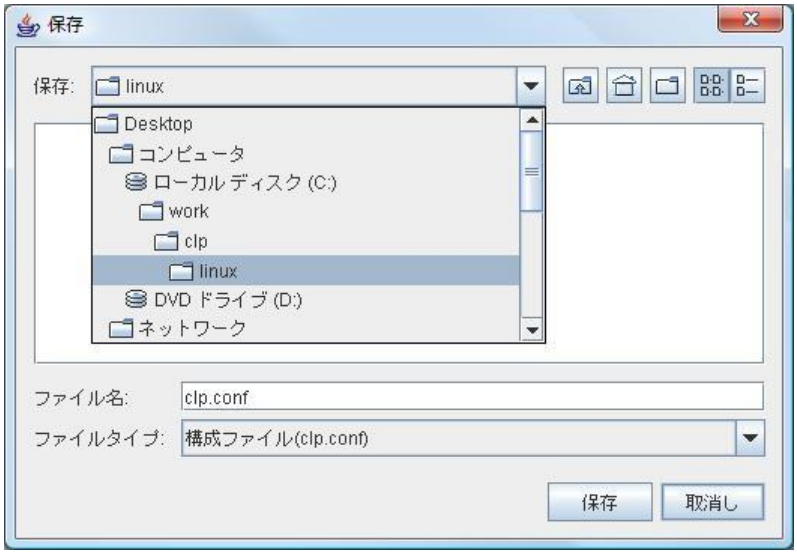
注 2: ファイル、ディレクトリが見えるのは、[Windows 用] または [ファイルシステム] を選択した場合のみです。

4. ファイルシステム内を参照し、ファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) が保存先のディレクトリ直下に作成されていることを確認します。

構成情報をファイル システムへ保存する (Windows)

Windows マシン使用時に、ファイル システムに構成情報を保存するには、以下の手順に従ってください。

1. Builder の [ファイル] メニューから [設定のエクスポート] をクリックするか、 ツールバーの [] をクリックします。
2. 以下のダイアログ ボックスで保存先を選択し、[保存] をクリックします。



3. 任意のディレクトリを選択または作成し、[保存] をクリックします。

注: 保存されるのはファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) です。これらのファイルとディレクトリがすべて揃っていない場合は生成コマンドの実行が不成功に終わりますので、移動する場合はかならずこの 2 点をセットとして取り扱ってください。なお、新規作成した構成情報を変更した場合は、上記 2 点に加えて clp.conf.bak が作成されます。

4. ファイルシステム内を参照し、ファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) が保存先のディレクトリ直下に作成されていることを確認します。

構成情報を FD に保存するには (Linux)

Linux マシン上の Builder で作成した構成情報を FD に保存するには、以下の手順に従ってください。

1. FD 装置に FD を挿入し、[ファイル] メニューの [設定のエクスポート] をクリックするか、ツールバーの [] をクリックします。
2. 以下のダイアログ ボックスが表示されます。FD のデバイス名を選択し、[OK] ボタンをクリックします。通常、FD の内部にディレクトリを作成せず、そのまま FD の直下に保存します。

注:

Windows 上のブラウザで動作する Builder でもこの構成情報を編集したい場合は、[Windows 用] チェック ボックスをオンにします。この場合は Windows で FAT(VFAT) フォーマットした 1.44MB の FD を用意します。

保存されるのはファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) です。これらのファイルとディレクトリがすべて揃っていない場合は生成コマンドの実行が不成功に終わりますので、移動する場合はかならずこの 2 点をセットとして取り扱ってください。なお、新規作成した構成情報を変更した場合は、上記 2 点に加えて clp.conf.bak が作成されます。

3. FD 内部を参照し、ファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) が FD の直下に作成されていることを確認します。

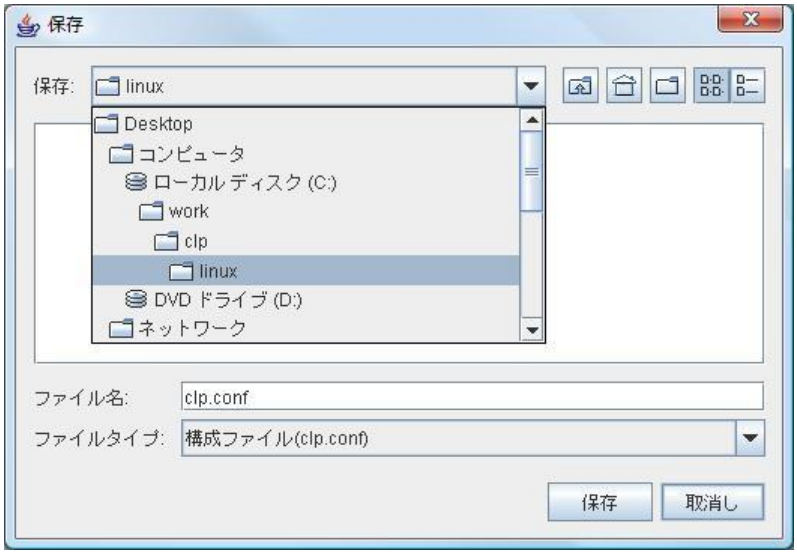
構成情報を FD に保存するには (Windows)

Windows マシン上の Builder で作成した構成情報を FD に保存するには、以下の手順に従ってください。

1 1.44MB のフォーマット済みの FD を用意します。

2 FD 装置に FD を挿入し、Builder の [ファイル] メニューの [設定のエクスポート] をクリックするか、ツールバーの [] をクリックします。通常、FD の内部にディレクトリを作成せず、そのまま FD の直下に保存します。

3 以下のダイアログ ボックスが表示されます。[保存] ボックスで FD のドライブを選択し、[保存] をクリックします。



注: Windows 上のブラウザで動作する Builder でもこのクラスタ構成情報を編集したい場合は、[Windows 用] チェック ボックスをオンにします。この場合は Windows で FAT(VFAT) フォーマットした 1.44MB の FD を用意します。

保存されるのはファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) です。これらのファイルとディレクトリがすべて揃っていない場合はクラスタ生成コマンドの実行が不成功に終わりますので、移動する場合はかならずこの 2 点をセットとして取り扱ってください。なお、新規作成した構成情報を変更した場合は、上記 2 点に加えて clp.conf.bak が作成されます。

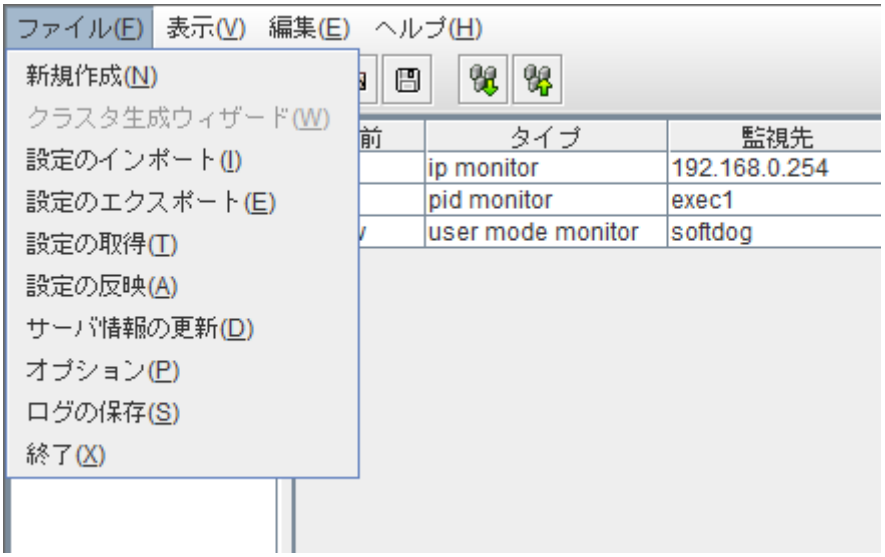
4 FD 内部を参照し、ファイル 1 点 (clp.conf) とディレクトリ 1 点 (scripts) が FD の直下に作成されていることを確認します。

構成情報を反映する

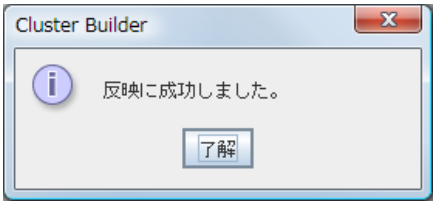
Builder(WebManager の設定モード)で構成情報を作成したら、サーバに構成情報を反映させます。

構成情報を反映するには、以下の手順に従ってください。

1. WebManager の設定モード(オンライン版 Builder)画面の [ファイル] メニューから、[設定の反映] をクリックします。



2. 設定の反映前後の構成情報の差異によっては、ポップアップウィンドウに反映に必要な動作に関する確認が表示されます。
   動作内容に問題がなければ、[OK]をクリックします。
3. 反映に成功すると、以下の画面が現れます。



注: 反映に失敗した場合は、表示されるメッセージに従って操作を行ってください。

オフライン版 Builder 利用時の差異について

オフライン版 Builder を利用する場合は、初期構築と構成情報の反映手順に違いがあります。

1. サーバの設定

1. Builder の[ファイル] メニューから、 [クラスタ生成ウィザード] をクリックします。[クラスタ生成ウィザード] ダイアログ ボックスが表示されます。[言語] フィールドには、WebManager を使用するマシンの OS で使用している言語を選択します。[次へ] をクリックします。
2. [名前] ボックスにサーバ名 (server1) を入力します。[次へ] をクリックします。

   テーブルビューは以下のようになります。

2. 構成情報を反映する

1. Web ブラウザを使用して、CLUSTERPRO Builder を起動します。
   (インストールパス)/clptrek.htm
2. 保存しておいた構成情報を開きます。
3. 構成情報が表示されるので、構成情報を変更します。
4. 変更した構成情報を保存します。
5. CLUSTERPRO Server がインストールされているサーバからコマンドプロンプトを利用して保存した構成情報を反映します。
   clpcfctrl –push -w –x <構成情報が保存されているパス>
   その際、変更した情報によってはサーバのサスペンドや停止、サーバシャットダウンによる再起動などが必要になります。このような場合は設定の反映が一旦キャンセルされ、必要な操作が表示されますので、表示されたメッセージにしたがって操作を行い、再度コマンドを実施してください。

第 3 章　　システムを確認する

本章では、作成したシステムが正常に起動するかどうかを確認します。

本章で説明する項目は以下のとおりです。

WebManager による動作確認

設定後のシステムの確認には、WebManager を使用して行う方法と、コマンドラインを使用して行う方法があります。本トピックでは、WebManager を使用してシステムの確認を行う方法について説明します。WebManager は、CLUSTERPRO Server のインストール時点ですでにインストールされています。新たにインストールを行う必要はありません。ここでは、まず WebManager の概要を説明し、その後、WebManager にアクセスし、サーバの状態を確認する方法について説明します。

関連情報: WebManager の動作環境については、『インストールガイド』の「第 1 章 CLUSTERPRO X SingleServerSafe の動作環境を確認する（ソフトウェア）」を参照してください。

WebManager を接続後、以下の手順で動作を確認します。

関連情報: WebManager の操作方法については『操作ガイド』の「第 1 章 WebManager の機能」を参照してください。

1. ハートビート リソース

   WebManager 上でサーバのステータスが ONLINE であることを確認します。
   サーバのハートビート リソースのステータスが NORMAL であることを確認します。

2. モニタ リソース

   WebManager 上で各モニタ リソースのステータスが NORMAL であることを確認します。

3. グループ起動

   グループを起動します。
   WebManager 上でグループのステータスが ONLINE であることを確認します。

4. EXEC リソース

   EXEC リソースを持つグループが起動しているサーバで、アプリケーションが動作していることを確認します。

5. グループ停止

   グループを停止します。
   WebManager 上でグループのステータスが OFFLINE であることを確認します。

6. グループ起動

   グループを起動します。
   WebManager 上でグループのステータスが起動済であることを確認します。

7. サーバ シャットダウン

   サーバをシャットダウンします。サーバが正常にシャットダウンされることを確認します。

コマンドによるサーバの動作確認

生成後、コマンドラインを使用して構成するサーバ上から状態を確認するには、以下の手順で動作を確認します。

関連情報: コマンドの操作方法については『操作ガイド』の「第 2 章 CLUSTERPRO X SingleServerSafe コマンド リファレンス」を参照してください。

clpstat コマンドを使用して、サーバのステータスが ONLINE であることを確認します。
サーバのハートビート リソースのステータスが NORMAL で あることを確認します。

1. ハートビート リソース

   clpstat コマンドを使用して、サーバのステータスが ONLINE であることを確認します。
   サーバのハートビート リソースのステータスが NORMAL であることを確認します。

2. モニタ リソース

   clpstat コマンドを使用して、各モニタ リソースのステータスが NORMAL であることを確認します。

3. グループ起動

   clpgrp コマンドを使用して、グループを起動します。
   clpstat コマンドを使用して、グループのステータスが ONLINE であることを確認します。

4. EXEC リソース

   EXEC リソースを持つグループが起動しているサーバで、アプリケーションが動作していることを確認します。

5. グループ停止

   clpgrp コマンドを使用して、グループを停止します。
   clpstat コマンドを使用して、グループのステータスが OFFLINE であることを確認します。

6. グループ起動

   clpgrp コマンドを使用して、グループを起動します。
   clpstat コマンドを使用して、グループのステータスが ONLINE であることを確認します。

7. シャットダウン

   clpstdn コマンドを使用してサーバをシャットダウンします。サーバが正常にシャットダウンされることを確認します。

セクション III　リソース詳細

このセクションでは、リソースについての詳細を説明します。CLUSTERPRO X SingleServerSafe は、クラスタリングソフトウェアである CLUSTERPRO X との操作性などにおける親和性を高めるために、共通の画面を使用しています。本ガイドでは、CLUSTERPRO X SingleServerSafeに特化した説明を行っていますので、設定項目の全体像を理解する際は、CLUSTERPRO X の『リファレンスガイド』を合わせて参照してください。

第 4 章　　グループリソースの詳細

本章では、グループリソースについての詳細を説明します。

本章で説明する項目は以下のとおりです。

グループリソース一覧

グループリソースとして定義可能なリソースは以下の通りです。

グループリソース名	機能	略称
EXECリソース	グループの起動時、終了時に実行されるアプリケーションやシェルスクリプトを登録します。	exec
仮想マシンリソース	仮想マシンの起動、停止を行います。	vm

仮想マシンリソースの動作環境

仮想マシンリソースの動作確認を行った仮想化基盤のバージョン情報を下記に提示します。

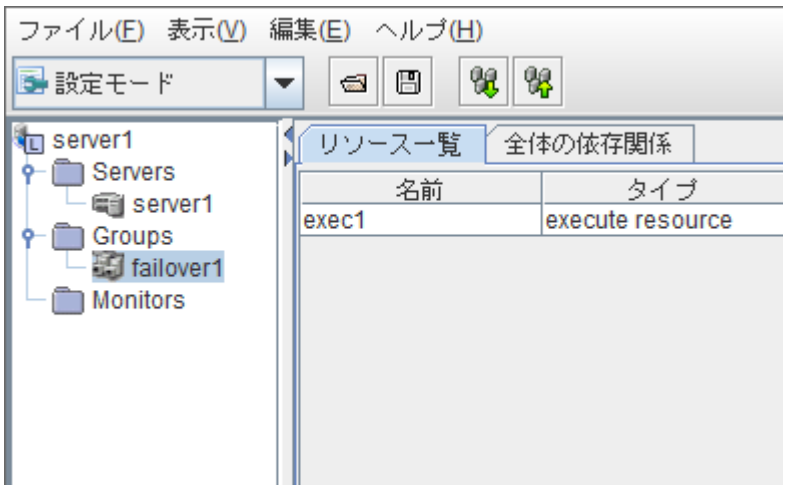
仮想化基盤	バージョン	CLUSTERPRO Version	備考
vSphere	4.0 update1	3.0.0-1~	x86_64
	4.0 update2	3.0.0-1~	x86_64
	4.1	3.0.0-1~	x86_64
	5	3.1.0-1~	VM
XenServer	5.5	3.0.0-1~	IA32
	5.6	3.0.0-1~	IA32
KVM	Red Hat Enterprise Linux 5.5	3.0.0-1~	x86_64
	Red Hat Enterprise Linux 5.6	3.0.0-1~	x86_64
	Red Hat Enterprise Linux 6.0	3.1.0-1~	x86_64
	Red Hat Enterprise Linux 6.1	3.1.0-1~	x86_64

EXEC リソースの設定

CLUSTERPRO では、CLUSTERPRO によって管理され、グループの起動時、終了時に実行されるアプリケーションやシェルスクリプトを登録できます。EXEC リソースには、ユーザ独自のプログラムやシェルスクリプトなども登録できます。シェルスクリプトは、sh のシェルスクリプトと同じ書式なので、それぞれのアプリケーションの事情にあわせた処理を記述できます。

1. ツリー ビューの [failover1] をクリックし、[編集] メニューの [追加] をクリックします。
2. [リソースの定義] ダイアログ ボックスが開きます。[タイプ] ボックスでグループ リソースのタイプ (execute resource) を選択し、[名前] ボックスにグループ名 (exec1) を入力します。[次へ] をクリックします。
3. 依存関係設定のページが表示されます。何も指定せず [次へ] をクリックします。
4. 復旧動作設定が表示されます。[次へ] をクリックします。
5. [ユーザアプリケーション] をチェックします。また、[Start path]に、実行ファイルのパスを指定します。[完了] をクリックします。

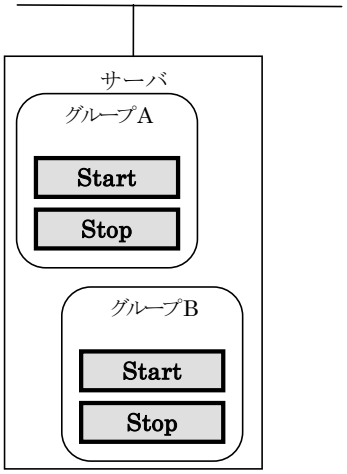
   テーブルビューは以下のようになります。



EXEC リソースで使用するスクリプト

スクリプトの種類

EXEC リソースには、それぞれ開始スクリプトと終了スクリプトが用意されています。CLUSTERPRO は、サーバの状態遷移が必要な場面において、EXEC リソースごとのスクリプトを実行します。動作させたいアプリケーションの起動、終了、もしくは復旧の手順を、これらのスクリプトに記述する必要があります。



Start　　開始スクリプト
Stop　　終了スクリプト

EXEC リソースのスクリプトで使用する環境変数

CLUSTERPRO は、スクリプトを実行する場合に、どの状態で実行したか(スクリプト実行要因)などの情報を環境変数にセットします。

スクリプト内で下図の環境変数を分岐条件として、システム運用にあった処理内容を記述できます。

終了スクリプトの環境変数は、直前に実行された開始スクリプトの内容を、値として返します。開始スクリプトでは CLP_FACTOR および CLP_PID の環境変数はセットされません。

CLP_LASTACTION の環境変数は、CLP_FACTOR の環境変数が CLUSTERSHUTDOWN または SERVERSHUTDOWN の場合にのみセットされます。

環境変数	環境変数の値	意味
CLP_EVENT …スクリプト実行要因	START	グループの起動により、実行された場合。 グループの移動により、移動先のサーバで実行された場合。 モニタリソースの異常検出によるグループの再起動により、同じサーバで実行された場合。 モニタリソースの異常検出によるグループリソースの再起動により、同じサーバで実行された場合。
	FAILOVER	使用しません。
CLP_FACTOR …グループ停止要因	CLUSTERSHUTDOWN	サーバ停止により、グループの停止が実行された場合。
	SERVERSHUTDOWN	サーバ停止により、グループの停止が実行された場合。
	GROUPSTOP	グループ停止により、グループの停止が実行された場合。
	GROUPMOVE	使用しません。
	GROUPFAILOVER	使用しません。
	GROUPRESTART	モニタリソースの異常検出により、グループの再起動が実行された場合。
	RESOURCERESTART	モニタリソースの異常検出により、グループリソースの再起動が実行された場合。
CLP_LASTACTION …停止後処理	REBOOT	OSをreboot (再起動) する場合。
	HALT	OSをhalt (シャットダウン) する場合。
	NONE	何もしない。
CLP_SERVER	HOME	使用しません。
	OTHER	使用しません。
CLP_DISK	SUCCESS	使用しません。
	FAILURE	使用しません。
CLP_PRIORITY	1～クラスタ内のサーバ数	使用しません。

環境変数	環境変数の値	意味
CLP_GROUPNAME …グループ名	グループ名	スクリプトが属している、グループ名を示す。
CLP_RESOURCENAME …リソース名	リソース名	スクリプトが属している、リソース名を示す。
CLP_PID …プロセスID	プロセスID	プロパティとして開始スクリプトが非同期に設定されている場合、開始スクリプトのプロセスIDを示す。開始スクリプトが同期に設定されている場合、本環境変数は値を持たない。
CLP_VERSION_FULL …CLUSTERPROフルバージョン	CLUSTERPROフルバージョン	CLUSTERPROのフルバージョンを示す。 （例）3.3.0-1
CLP_VERSION_MAJOR …CLUSTERPROメジャーバージョン	CLUSTERPROメジャーバージョン	CLUSTERPROのメジャーバージョンを示す。 （例）3
CLP_PATH …CLUSTERPROインストールパス	CLUSTERPROインストールパス	CLUSTERPROがインストールされているパスを示す。 （例）/opt/nec/clusterpro
CLP_OSNAME …サーバOS名	サーバOS名	スクリプトが実行されたサーバのOS名を示す。 （例） ①OS名が取得できた場合： Red Hat Enterprise Linux Server release 6.0 (Santiago) ②OS名が取得できなかった場合： Linux
CLP_OSVER …サーバOSバージョン	サーバOSバージョン	スクリプトが実行されたサーバのOSバージョンを示す。 （例） ①OSバージョンが取得できた場合：6.0 ②OSバージョンが取得できなかった場合：※値なし

EXEC リソース スクリプトの実行タイミング

開始、終了スクリプトの実行タイミングと環境変数の関連を、状態遷移図にあわせて説明します。

- 図中の○や×はサーバの状態を表しています。

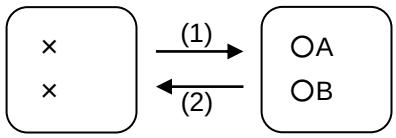
サーバ	サーバ状態
○	正常状態
×	停止状態

(例)○A ： 正常状態にあるサーバにおいてグループ A が動作している。

- 定義されているグループはA、Bの2つ。

【状態遷移図】

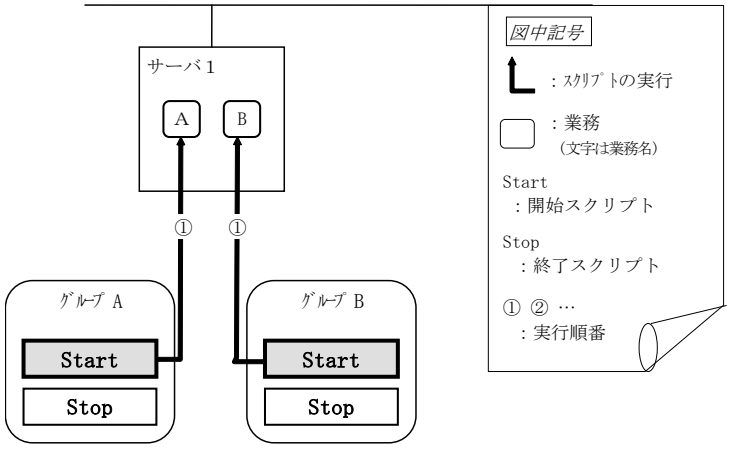
状態遷移について説明します。



図中の(1)～(2)は、以下の説明に対応しています。

(1) 通常立ち上げ

ここでいう通常立ち上げとは、開始スクリプトがサーバで正常に実行された時を指します。

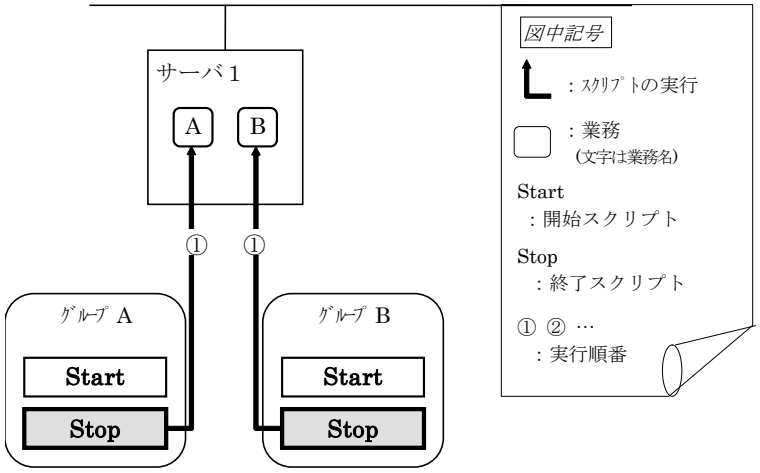


Start に対する環境変数

グループ	環境変数	値
A	CLP_EVENT	START
B	CLP_EVENT	START

(2) 通常シャットダウン

ここでいう通常シャットダウンとは、終了スクリプトに対応する開始スクリプトが、通常立ち上げにより実行されたシャットダウンを指します。



Stop に対する環境変数

グループ	環境変数	値
A	CLP_EVENT	START
B	CLP_EVENT	START

EXEC リソーススクリプトの記述の流れ

前のトピックの、スクリプトの実行タイミングと実際のスクリプト記述を関連付けて説明します。文中の(数字)は 61 ページの「EXEC リソース スクリプトの実行タイミング」の各動作をさします。

グループ A 開始スクリプト：start.sh の一例

```
#! /bin/sh
# ****************************************
# *                start.sh                *
# ****************************************

if [ "$CLP_EVENT" = "START" ]
then
        処理概要：
                業務の通常起動処理
        この処理を行う実行タイミング：
                (1) 通常立ち上げ

else
        #NO_CLP

fi
#EXIT

exit 0
```

スクリプト実行要因の環境変数を参照して、処理の振り分けを行う。

CLUSTERPROは動作していない

終了コードが0の場合、EXECリソースの活性処理は成功と判定される。
スクリプト内でエラーが発生した場合には0以外の終了コードを返却するように記述する。

グループ A 終了スクリプト：stop.sh の一例

```
#! /bin/sh
# ***************************************
# *                stop.sh                *
# ***************************************

if [ "$CLP_EVENT" = "START" ]
then
        処理概要：
                業務の通常終了処理
        この処理を行う実行タイミング：
                (2) 通常シャットダウン




else
        #NO_CLP

fi
#EXIT
exit 0
```

スクリプト実行要因の環境変数を参照して、処理の振り分けを行う。

CLUSTERPROは動作していない

EXEC リソーススクリプト作成のヒント

以下の点に注意して、スクリプトを作成してください。

- スクリプト中にて、実行に時間を必要とするコマンドを実行する場合には、コマンドの実行が完了したことを示すトレースを残すようにしてください。この情報は、問題発生時、障害の切り分けを行う場合に使用することができます。トレースを残す方法は下記の2つがあります。
- スクリプト中にechoコマンドを記述してEXECリソースのログ出力先を設定する方法
  トレースをechoコマンドにて標準出力することができます。その上で、スクリプトが属しているリソースのプロパティでログ出力先を設定します。

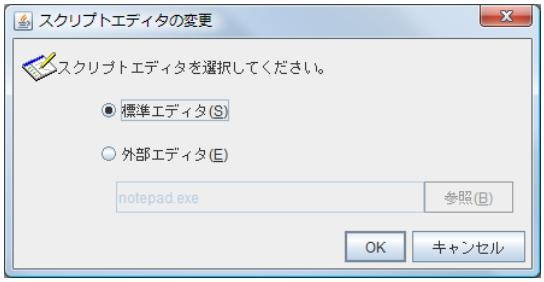
  デフォルトではログ出力されません。ログ出力先の設定については 74 ページの「EXEC リソースの調整を行うには」を参照してください。[ローテートする]チェックボックスがオフの場合は、ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意してください。

  (例:スクリプト中のイメージ)
  ```
  echo "appstart.."
  appstart
  echo "OK"
  ```
- スクリプト中にclplogcmdを記述する方法
  clplogcmdでWebManager のアラートビューやOSのsyslogに、メッセージを出力できます。clplogcmdについては、「操作ガイド」の「第 2 章 CLUSTERPROSingleServerSafe コマンドリファレンス」の「メッセージ出力コマンド」を参照してください。

  (例:スクリプト中のイメージ)
  ```
  clplogcmd -m "appstart.."
  appstart
  clplogcmd -m "OK"
  ```

変更

スクリプトエディタの変更ダイアログが表示されます。スクリプトを表示または編集するエディタを任意のエディタに変更できます。



標準エディタ

スクリプトエディタに標準のエディタを使用します。

- Linux … vi(実行ユーザのサーチパスで検索される vi)
- Windows … メモ帳(実行ユーザのサーチパスで検索される notepad.exe)

外部エディタ

スクリプトエディタを任意に指定します。[参照]を選択し、使用するエディタを指定します。

Linux で CUI ベースのエディタを外部エディタで指定するにはシェルスクリプトを作成してください。

以下の例は vi を実行するシェルスクリプトです。

xterm -name clpedit -title "Cluster Builder" -n "Cluster Builder" -e vi "$1"

調整

EXEC リソース調整プロパティダイアログを表示します。EXEC リソースの詳細設定を行います。EXEC リソースを PID モニタリソースで監視するには、開始スクリプトの設定を非同期にする必要があります。

EXEC リソース 注意事項

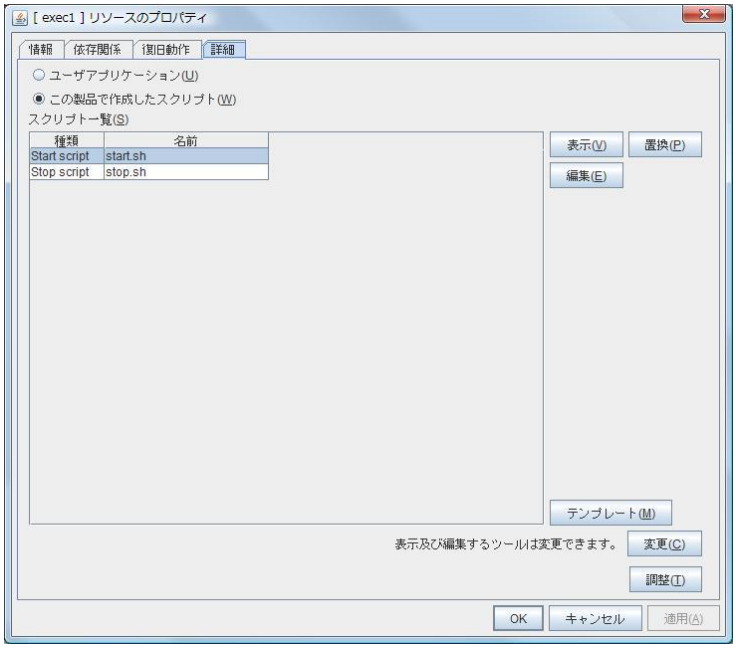
- スクリプトのログローテート機能について

  スクリプトのログローテート機能を有効にした場合、スクリプト終了時に指定されたファイルへの書き込みが行われます。開始スクリプトを [非同期] に設定している場合、スクリプトが終了することなく常駐し、ログをリアルタイムで確認することができないため、ログローテート機能を無効にすることを推奨します。開始スクリプトを [同期] に設定している場合、開始スクリプトから起動した常駐プロセスの標準出力を /dev/null へ指定してください。

- 開始スクリプト/終了スクリプトは root ユーザで実行されます。
- 環境変数に依存するアプリケーションを起動する際には必要に応じてスクリプト側で環境変数の設定を行っていただく必要があります。

EXEC リソースの詳細を表示 / 変更するには

1. Builder の左部分に表示されているツリービューから、詳細情報の表示 / 設定変更を行いたい EXEC リソースが所属するグループのアイコンをクリックします。
2. 画面右のテーブルビューに、グループリソースの一覧が表示されます。目的の EXEC リソース名を右クリックし、[プロパティ] の [詳細] タブをクリックします。
3. 以下の説明にしたがって、詳細設定の表示 / 変更を行います。



ユーザアプリケーション

スクリプトとしてサーバ上の実行可能ファイル(実行可能なシェルスクリプトやバイナリファイル)を使用します。各実行可能ファイル名は、サーバ上のローカルディスクのパスで設定します。

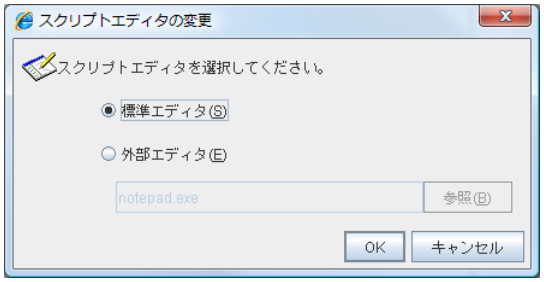
Builder の構成情報には含まれません。スクリプトファイルは Builder では編集できません。

この製品で作成したスクリプト

スクリプトとして Builder で準備したスクリプトファイルを使用します。必要に応じて Builder でスクリプトファイルを編集できます。スクリプトファイルは、構成情報に含まれます。

変更

スクリプトエディタの変更ダイアログが表示されます。スクリプトを表示または編集するエディタを任意のエディタに変更できます。



標準エディタ

スクリプトエディタに標準のエディタを使用します。

- Linux … vi(実行ユーザのサーチパスで検索される vi)
- Windows … メモ帳(実行ユーザのサーチパスで検索される notepad.exe)

外部エディタ

スクリプトエディタを任意に指定します。[参照]を選択し、使用するエディタを指定します。

Linux で CUI ベースのエディタを外部エディタで指定するにはシェルスクリプトを作成してください。

以下の例は vi を実行するシェルスクリプトです。

xterm -name clpedit -title "Cluster Builder" -n "Cluster Builder" -e vi "$1"

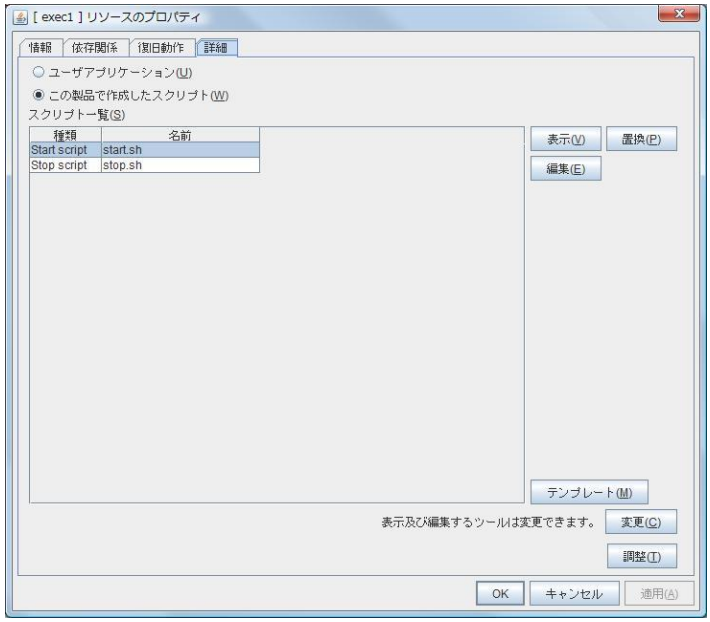
調整

EXEC リソース調整プロパティダイアログを表示します。EXEC リソースの詳細設定を行います。EXEC リソースを PID モニタリソースで監視するには、開始スクリプトの設定を非同期にする必要があります。

Builder で作成した EXEC リソース スクリプトを表示 / 変更するには

1. Builder の左部分に表示されているツリービューから、詳細情報の表示 / 設定変更を行いたい EXEC リソースが所属するグループのアイコンをクリックします。
2. 画面右のテーブルビューに、グループリソースの一覧が表示されます。目的の EXEC リソース名を右クリックし、[プロパティ] の [詳細] タブをクリックします。
3. [詳細] タブで、[この製品で作成したスクリプト] をクリックします。
4. 以下の説明に従い詳細設定の表示 / 変更を行います。

   [スクリプト一覧] に既定のスクリプトファイル名 [start.sh]、[stop.sh] が表示されます。



表示

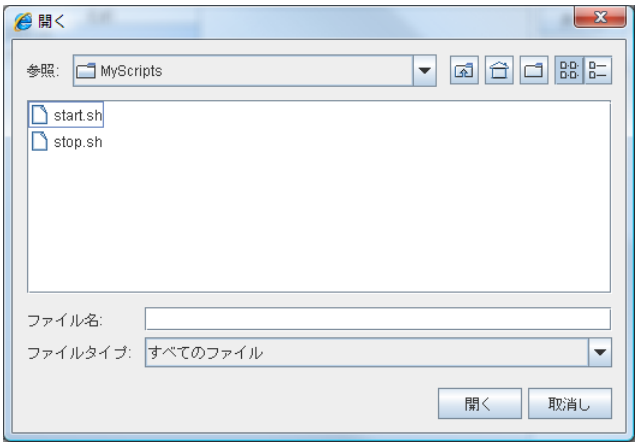
選択したスクリプトファイルをスクリプトエディタで表示します。エディタで編集して保存した内容は反映されません。表示しようとしているスクリプトファイルが表示中または編集中の場合は表示できません。

編集

選択したスクリプトファイルをスクリプトエディタで編集できます。変更を反映するには上書き保存を実行してください。編集しようとしているスクリプトファイルが表示中または編集中の場合は編集できません。スクリプトファイル名の変更はできません。

置換

ファイル選択ダイアログ ボックスが表示されます。



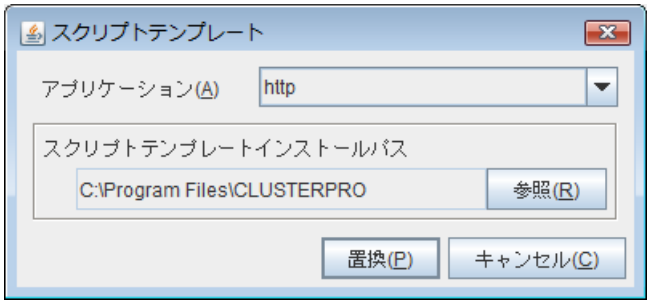
[リソースのプロパティ] で選択したスクリプトファイルの内容が、ファイル選択ダイアログボックスで選択したスクリプトファイルの内容に置換されます。スクリプトが表示中または編集中の場合は置換できません。ここではスクリプトファイルを選択してください。バイナリファイル(アプリケーションなど)は選択しないでください。

スクリプトテンプレートの簡易選択機能を利用するには

EXEC リソースでアプリケーションを選択することにより、必要なスクリプトテンプレートを自動置換することができます。そのスクリプトを編集することで容易にスクリプトを作成することができます。

注:本機能を利用するためには、事前にスクリプトテンプレートをインストールしておく必要があります。

1. Builder の左部分に表示されているツリービューから、スクリプトテンプレートの置換を行いたい EXEC リソースが所属するグループのアイコンをクリックします。
2. 画面右のテーブルビューに、グループリソースの一覧が表示されます。目的の EXEC リソース名を右クリックし、[プロパティ] の [詳細] タブをクリックします。
3. [詳細] タブで、[この製品で作成したスクリプト] をクリックします。
4. [テンプレート] ボタンをクリックします。
5. [スクリプトテンプレート] ダイアログが表示されます。



アプリケーション

置換可能なスクリプトテンプレートのアプリケーション一覧がリストボックスで表示されます。

注:スクリプトテンプレートがインストールされていない場合、アプリケーション一覧には何も表示されません。

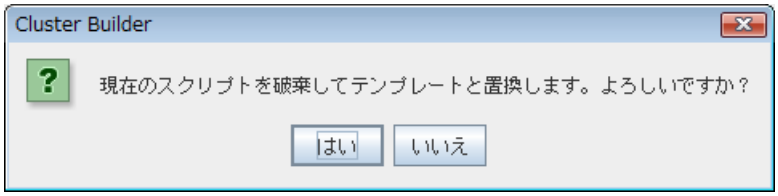
参照

スクリプトテンプレートがインストールされているフォルダパスを参照します。

注:既定のフォルダパスにスクリプトテンプレートがインストールされていない場合、警告メッセージが表示されます。スクリプトテンプレートをインストールしている場合は正しいインストールパスを指定してください。

置換

スプリプト置換確認のダイアログボックスが表示されます。



[OK] ボタンをクリックした場合、スクリプトの置換が実行されます。

注: 置換したスクリプトは環境に応じて編集する必要があります。スクリプトの編集方法については、「Builder で作成した EXEC リソース スクリプトを表示 / 変更するには」を参照してください。

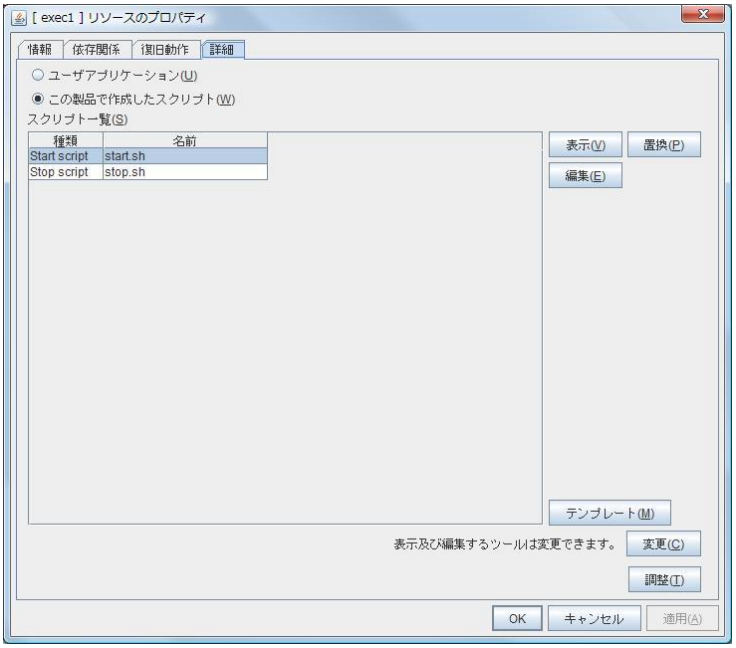
ユーザアプリケーションを使用した EXEC リソース スクリプトを表示 / 変更するには

1. Builder の左部分に表示されているツリービューから、詳細情報の表示 / 設定変更を行いたい EXEC リソースが所属するグループのアイコンをクリックします。
2. 画面右のテーブルビューに、グループリソースの一覧が表示されます。目的の EXEC リソース名を右クリックし、[プロパティ] の [詳細] タブをクリックします。
3. [詳細] タブで、[ユーザアプリケーション] をクリックします。
4. 以下の説明に従い詳細設定の表示 / 変更を行います。

   EXEC リソースの実行可能ファイルとして任意のファイルを設定します。[スクリプト一覧] には設定した実行可能ファイル名が表示されます。

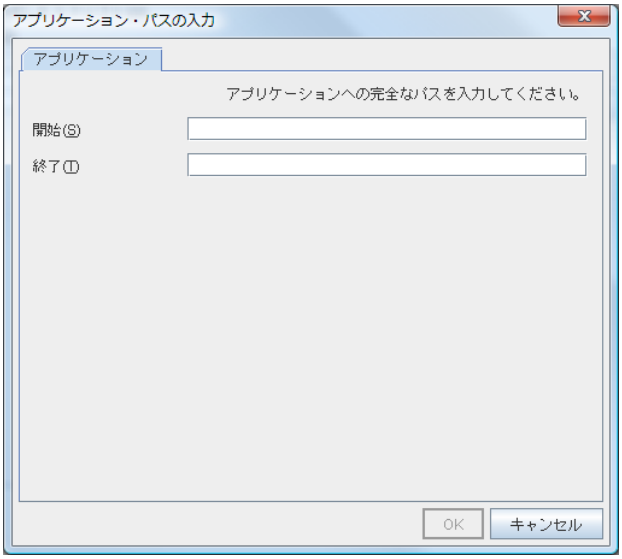
   実行可能ファイルとは実行可能なシェルスクリプトやバイナリファイルです。

   Linux で動作する Builder に設定されている標準のスクリプトエディタは vi です。表示および編集のウィンドウを閉じる場合は、vi の q コマンドで終了してください。



編集

EXEC リソースの実行可能ファイル名を設定します。アプリケーション パスの入力ダイアログ ボックスが表示されます。



開始 (1023 バイト以内)

EXEC リソースの開始時の実行可能ファイル名を設定します。「/」で始まる必要があります。引数を指定することも可能です。

終了(1023 バイト以内)

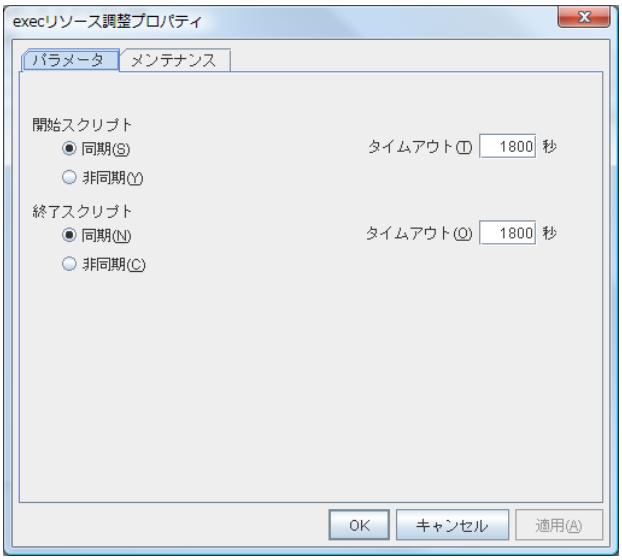
EXEC リソースの終了時の実行可能ファイル名を設定します。「/」で始まる必要があります。終了スクリプトは省略可能です。

実行可能ファイル名はサーバ上のファイルを「/」から始まる完全なパス名で設定する必要があります。引数を指定することも可能です。

EXECリソースの調整を行うには

1. Builder の左部分に表示されているツリービューから、詳細情報の表示 / 設定変更を行いたい EXEC リソースが所属するグループのアイコンをクリックします。
2. 画面右のテーブルビューに、グループリソースの一覧が表示されます。目的の EXEC リソース名を右クリックし、[プロパティ] の [詳細] タブをクリックします。
3. [詳細] タブで、[調整] をクリックします。[exec リソース調整プロパティ] ダイアログ ボックスが表示されます。
4. 以下の説明に従い詳細設定の表示 / 変更を行います。

パラメータタブ



[開始スクリプト]、[終了スクリプト]全スクリプト共通

同期

スクリプトの実行時にスクリプトの終了を待ちます。常駐しない(実行後に処理がすぐ戻る)実行可能ファイルの場合に選択します。

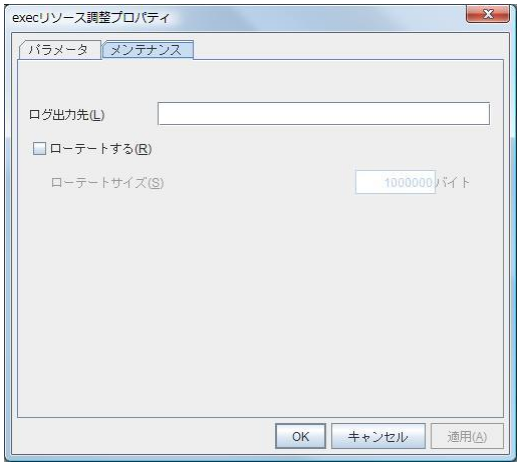
非同期

スクリプトの実行時にスクリプトの終了を待ちません。常駐する実行可能ファイルの場合に選択します。
EXECリソースの開始スクリプトを非同期で実行する場合は、PIDモニタリソースで監視できます。

タイムアウト(1～9999)

スクリプトの実行時に終了を待つ場合([同期])のタイムアウトを設定します。[同期]を選択している場合のみ入力可能です。設定時間内にスクリプトが終了しないと、異常と判断します。

メンテナンスタブ



ログ出力先 (1023 バイト以内)

EXEC リソースのスクリプトや実行可能ファイルの標準出力と標準エラー出力のリダイレクト先を指定します。何も指定しない場合、/dev/null に出力されます。[/] で始まる必要があります。

[ローテートする]チェックボックスがオフの場合は無制限に出力されますのでファイルシステムの空き容量に注意してください。

i686 版を利用している場合、ファイルサイズが 2GB を超えると EXEC リソースの活性・非活性が行えなくなるので、ファイルを定期的に削除する必要があります。

[ローテートする]チェックボックスがオンの場合は、出力されるログファイルは、ローテートします。また、以下の注意事項があります。

[ログの出力先] には 1009 バイト以内でログのパスを記述してください。1010 バイトを超えた場合、ログの出力が行えません。

ログファイルの名前の長さは 31 バイト以内で記述してください。32 バイト以上の場合、ログの出力が行えません。

複数のカスタムモニタリソースでログローテートを行う場合、パス名が異なってもログファイルの名前が同じ場合、(ex. /home/foo01/log/genw.log, /home/foo02/log/genw.log) ローテートサイズが正しく反映されないことがあります。

ローテートする

EXEC リソースのスクリプトや実行可能ファイルの実行ログを、オフの場合は無制限のファイルサイズで、オンの場合はローテートして出力します。

ローテートサイズ (1～999999999)

[ローテートする]チェックボックスがオンの場合に、ローテートするサイズを指定します。

ローテート出力されるログファイルの構成は、以下のとおりです。

ファイル名	内容
[ログ出力先]指定のファイル名	最新のログです。
[ログ出力先]指定のファイル名.pre	ローテートされた以前のログです。

仮想マシンリソースの設定

仮想マシンリソースの依存関係

既定値では、依存するグループリソースタイプはありません。

仮想マシンリソースとは?

仮想化基盤のホスト OS 上から仮想マシン(ゲスト OS)の制御を行うためのリソースです。

仮想マシンの起動、停止を行います。

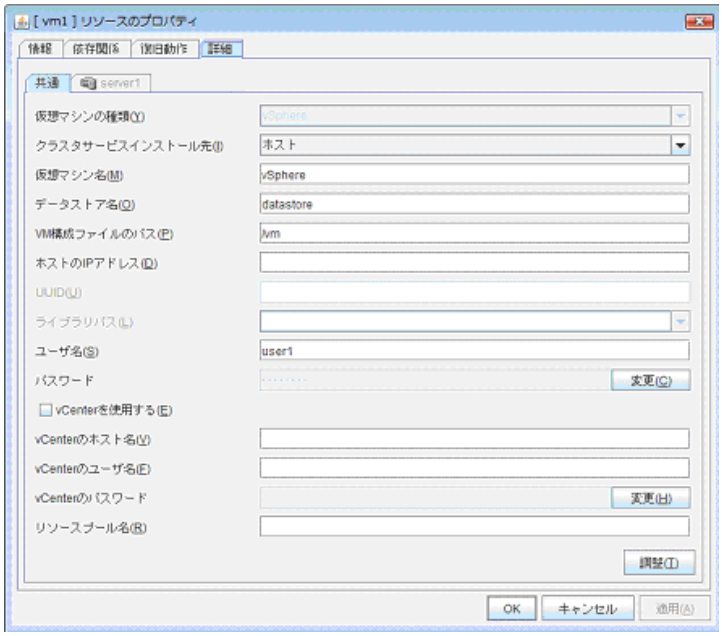
仮想マシンリソースに関する注意事項

- 仮想マシンリソースは CLUSTERPRO を仮想化基盤(vSphere, XenServer, KVM)のホスト OS 上にインストールした場合のみ有効です。
- 仮想マシン リソースはグループタイプが仮想マシンのグループにのみ登録可能です。
- 仮想マシン リソースは一つのグループに一つのみ登録可能です

仮想マシンリソースの詳細を表示 / 変更するには

1. Builder の左部分に表示されているツリービューから、詳細情報の表示 / 設定変更を行いたい仮想マシンリソースが所属するグループのアイコンをクリックします。
2. 画面右のテーブルビューに、グループリソースの一覧が表示されます。目的の仮想マシンリソース名を右クリックし、[プロパティ] の [詳細] タブをクリックします。
3. [詳細] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。

リソース詳細タブ(vSphere の場合)



仮想マシンの種類

仮想化基盤の種類を指定します。

クラスタサービスインストール先

CLUSTERPRO をインストールする OS の種類を指定します。ゲスト OS を選択すると、vCenter を使用するのチェックボックスも自動的にオンになります。

仮想マシン名(255 バイト以内)

仮想マシン名を入力してください。VM 構成ファイルのパスを入力する場合は設定不要です。また、仮想化基盤側で仮想マシン名を変更する可能性がある場合、VM 構成ファイルのパス名を設定してください。

データストア名 (255 バイト以内)

仮想マシンの設定情報を格納しているデータストア名を指定してください。

VM 構成ファイルのパス(1023 バイト以内)

仮想マシンの設定情報を格納しているパスを指定してください。

ホストの IP アドレス

ホストの管理 IP アドレスを指定してください。サーバ別設定を利用して、サーバごとに指定する必要があります。

ユーザ名(255 バイト以内)

仮想マシンを起動するために利用するユーザ名を指定してください。

パスワード(255 バイト以内)

仮想マシンを起動するために利用するパスワードを指定してください。

vCenter を使用する

vCenter を使用するかどうかを指定してください。

vCenter のホスト名(1023 バイト以内)

vCenter のホスト名を指定してください。

vCenter のユーザ名(255 バイト以内)

vCenter に接続するためのユーザ名を指定してください。

vCenter のパスワード(255 バイト以内)

vCenter に接続するためのパスワードを指定してください。

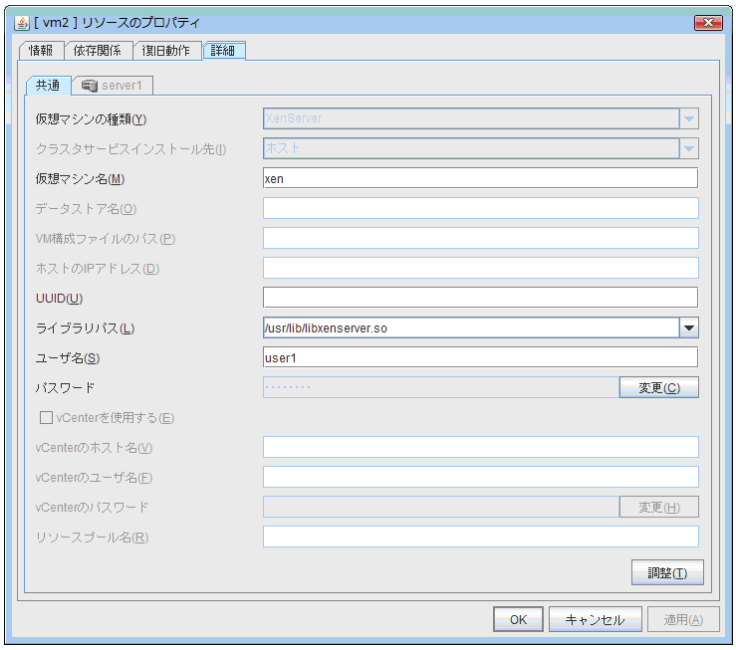
リソースプール名(80 バイト以内)

仮想マシンを起動するリソースプール名を指定します。

調整

[仮想マシンリソース調整プロパティ] ダイアログ ボックスを表示します。仮想マシンリソースの詳細設定を行います。

リソース詳細タブ(XenServer の場合)



仮想マシンの種類(255 バイト以内)

仮想化基盤の種類を指定します。

仮想マシン名(255 バイト以内)

仮想マシン名を入力してください。UUID を設定する場合は不要です。また、仮想化基盤側で仮想マシン名を変更する可能性がある場合、UUID を設定してください。

UUID

仮想マシンを識別するための UUID(Universally Unique Identifier)を指定してください。

ライブラリパス(1023 バイト以内)

XenServer を制御するために利用するライブラリのパスを指定してください。

ユーザ名(255 バイト以内)

仮想マシンを起動するために利用するユーザ名を指定してください。

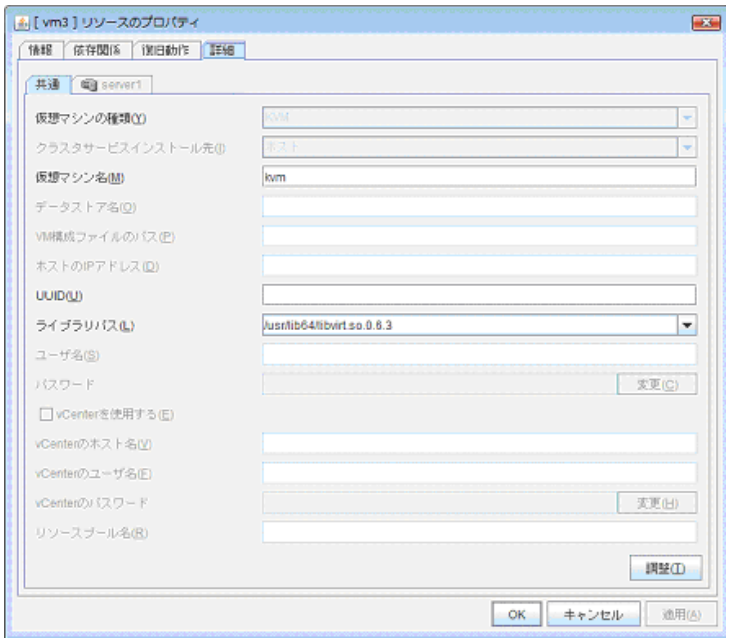
パスワード(255 バイト以内)

仮想マシンを起動するために利用するパスワードを指定してください。

調整

[仮想マシンリソース調整プロパティ] ダイアログ ボックスを表示します。仮想マシンリソースの詳細設定を行います。

リソース詳細タブ(KVM の場合)



仮想マシンの種類(255 バイト以内)

仮想化基盤の種類を指定します。

仮想マシン名(255 バイト以内)

仮想マシン名を入力してください。UUID を設定する場合は不要です。

UUID

仮想マシンを識別するための UUID(Universally Unique Identifier)を指定してください。

ライブラリパス(1023 バイト以内)

KVM を制御するために利用するライブラリのパスを指定してください。

調整

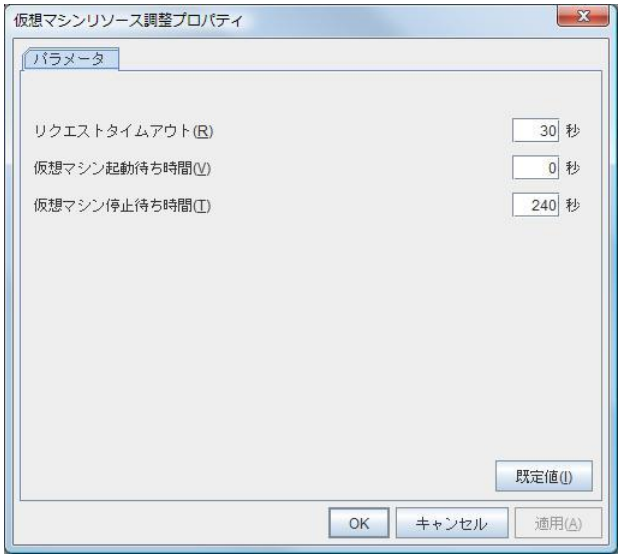
[仮想マシンリソース調整プロパティ] ダイアログ ボックスを表示します。仮想マシンリソースの詳細設定を行います。

仮想マシンリソースの調整を行うには

1. 仮想マシンリソースタブ場面で[調整」をクリックします。

2. 仮想マシンリソース調整プロパティの画面を表示します。以下の説明に従い詳細設定の表示 / 変更を行います。

仮想マシンリソース調整プロパティ



リクエストタイムアウト

仮想マシンの起動/停止などの要求の完了を待ち合わせる時間を指定します。

この時間内に要求が完了しなかった場合、タイムアウトと見なし、リソースの活性または非活性は失敗します。

仮想マシン起動待ち時間

仮想マシンの起動要求を発行した後で、この時間だけ必ず待ちます。

仮想マシン停止待ち時間

仮想マシンの停止を待ち合わせる最大の時間です。仮想マシンの停止が確認できた時点で非活性完了になります。

第 5 章　　モニタリソースの詳細

本章では、CLUSTERPRO X SingleServerSafe で監視を実行する単位であるモニタリソースについての詳細を説明します。

本章で説明する項目は以下のとおりです。

モニタリソース一覧

モニタリソースとして定義可能なリソースは以下の通りです。

モニタリソース名	機能	監視タイミング (太字は既定値)	対象リソース
ディスクモニタリソース	ディスクデバイスの監視を行います。	**常時**/活性時	全て
IPモニタリソース	ping コマンドを使用して応答の有無により、IPアドレスおよび通信路の監視を行います。	**常時**/活性時	全て
NIC Link Up/Downモニタリソース	NICのLink状態を取得し、LinkのUp/Downの監視を行います。	**常時**/活性時	全て
PIDモニタリソース	活性に成功したEXECリソースを監視します。	**常時**/活性時	全て
ユーザ空間モニタリソース	ユーザ空間のストールを異常として判断します。	常時(固定)	-
マルチターゲットモニタリソース	複数のモニタリソースの状態の組み合わせで監視を行います。	活性時(固定)	全て
ソフトRAIDモニタリソース	ソフトRAIDを行っているデバイスを監視します。	常時(固定)	なし
カスタムモニタリソース	任意のスクリプトを実行することで監視を行います。	**常時**/活性時	全て
ボリュームマネージャモニタリソース	複数のストレージやディスクの監視機構を提供します。	常時/**活性時**	全て
仮想マシンモニタリソース	仮想マシンリソースで起動した仮想マシンの監視機構を提供します。	常時(固定)	vm
外部連携モニタリソース	”異常発生通知受信時に実行する異常時動作の設定”と”異常発生通知のWebManager表示” を実現します。	常時 (固定)	なし
プロセス名モニタリソース	任意のプロセス名のプロセスを監視します。	**常時**/活性時	全て
DB2モニタリソース	IBM DB2データベースへの監視機構を提供します。	活性時(固定)	全て
FTPモニタリソース	FTPサーバへの監視	活性時(固定)	全て

	機構を提供します。		
HTTPモニタリソース	HTTPサーバへの監視機構を提供します。	活性時(固定)	全て
IMAP4モニタリソース	IMAPサーバへの監視機構を提供します。	活性時(固定)	全て
MySQLモニタリソース	MySQLデータベースへの監視機構を提供します。	活性時 (固定)	全て
NFSモニタリソース	NFSのファイルサーバへの監視機構を提供します。	**常時**/活性時	全て
Oracleモニタリソース	Oracleデータベースへの監視機構を提供します。	活性時(固定)	全て
Oracleアプリケーションサーバモニタリソース	Oracleアプリケーションサーバへの監視機構を提供します。	活性時(固定)	全て
POP3モニタリソース	POPサーバへの監視機構を提供します。	活性時(固定)	全て
PostgreSQLモニタリソース	PostgreSQLデータベースへの監視機構を提供します。	活性時(固定)	全て
Sambaモニタリソース	sambaファイルサーバへの監視機構を提供します。	**常時**/活性時	全て
SMTPモニタリソース	SMTPサーバへの監視機構を提供します。	活性時(固定)	全て
Sybaseモニタリソース	Sybaseデータベースへの監視機構を提供します。	活性時(固定)	全て
Tuxedoモニタリソース	Tuxedoアプリケーションサーバへの監視機構を提供します。	活性時(固定)	全て
Weblogicモニタリソース	WebLogicアプリケーションサーバへの監視機構を提供します。	活性時(固定)	全て
Websphereモニタリソース	WebSphereアプリケーションサーバへの監視機構を提供します。	活性時(固定)	全て
WebOTXモニタリソース	WebOTXアプリケーションサーバへの監視機構を提供します。	活性時(固定)	全て
JVMモニタリソース	Java VMの監視を行います。	**常時**/活性時	exec
システムモニタリソース	システムリソースの監視を行います。	常時 (固定)	全て

モニタリソースの監視開始後のステータス

モニタリソースの監視開始後、監視開始準備のために一時的にステータスが警告となることがあります。

モニタステータスが警告となる可能性があるのは、下記のモニタリソースです。

- 外部連携モニタリソース
- カスタムモニタリソース (監視タイプが [非同期] の場合のみ)
- DB2 モニタリソース
- システムモニタリソース
- JVM モニタリソース
- MySQL モニタリソース
- Oracle モニタリソース
- PostgreSQL モニタリソース
- プロセス名モニタリソース
- Sybase モニタリソース

モニタリソースの監視タイミング

モニタリソースによる監視は、常時監視と活性時監視の 2 つのタイプがあります。

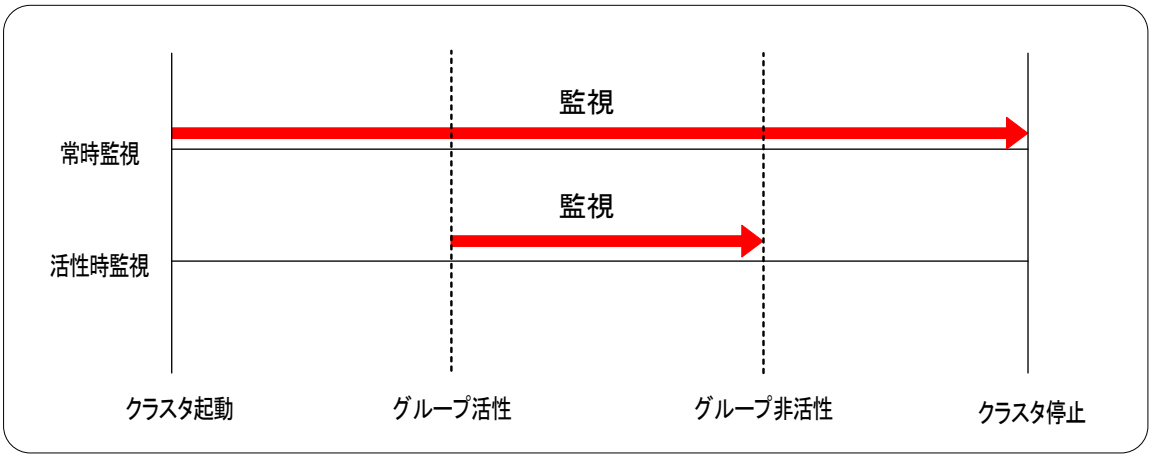
モニタリソースによって設定可能な監視タイミングが異なります。

- 常時

  モニタリソースは常に監視を行います。

- 活性時

  特定のグループリソースが活性状態の間、監視を実行します。グループリソースが非活性状態の間は監視を実行しません。



モニタリソースの一時停止/再開

モニタリソースは一時的に監視を停止したり再開したりすることが可能です。
監視の一時停止/再開の方法は以下の 2 つの方法があります。

- WebManager による操作
- clpmonctrl コマンドによる操作
  clpmonctrl コマンドでは、コマンドの実行サーバ上のモニタリソースの制御のみ可能です

モニタリソースには、一時停止/再開の制御が可能なものと不可能なものがあります。
モニタリソースによる制御の可否は下記を参照してください。

モニタリソース	制御可否
ディスクモニタリソース	可能
IPモニタリソース	可能
ユーザ空間モニタリソース	可能
NIC Link Up/Downモニタリソース	可能
PIDモニタリソース	可能
マルチターゲットモニタリソース	可能
カスタムモニタリソース	可能

ボリュームマネージャモニタリソース	可能
ソフトRAIDモニタリソース	可能
プロセス名モニタリソース	可能
DB2モニタリソース	可能
FTPモニタリソース	可能
HTTPモニタリソース	可能
IMAP4モニタリソース	可能
MySQLモニタリソース	可能
NFSモニタリソース	可能
Oracleモニタリソース	可能
OracleASモニタリソース	可能
POP3モニタリソース	可能
PostgreSQLモニタリソース	可能
Sambaモニタリソース	可能
SMTPモニタリソース	可能
Sybaseモニタリソース	可能
Tuxedoモニタリソース	可能
Weblogicモニタリソース	可能
Websphereモニタリソース	可能
WebOTXモニタリソース	可能
仮想マシンモニタリソース	可能
外部連携モニタリソース	可能
JVMモニタリソース	可能
システムモニタリソース	可能

WebManager では、制御が不可能なモニタリソースの右クリックメニューが無効になります。clpmonctrl コマンドでは、制御可能なモニタリソースのみの制御が行われます。制御が不可能なモニタリソースは警告メッセージが表示され制御は行われません。

モニタリソースが一時停止状態で下記の操作を行った場合、モニタリソースの一時停止が解除されます。

- WebManager で、モニタリソースの「再開」を行った場合
- clpmonctrl コマンドに –r オプション を指定した場合
- クラスタを停止した場合
- クラスタをサスペンドした場合

モニタリソースの擬似障害 発生/解除

モニタリソースは擬似的に障害を発生させることが可能です。また、それを解除することもできます。擬似障害の発生/解除を行う方法は以下の 2 つの方法があります。

- WebManager (検証モード) による操作
  WebManager (検証モード) では、制御が不可能なモニタリソースの右クリックメニューが無効になります。
- [clpmonctrl] コマンドによる操作
  [clpmonctrl] コマンドでは、コマンドを実行するサーバ上のモニタリソースに対して制御を行います。制御が不可能なモニタリソースに対して実行した場合、コマンドの実行自体は成功しますが、疑似障害を発生させることはできません。

モニタリソースには、擬似障害の発生/解除が可能なものと不可能なものがあります。
『操作ガイド』の『第 2 章 CLUSTERPRO X SingleServerSafe コマンドリファレンス 監視リソースを制御する (clpmonctrl コマンド)』を参照してください。

擬似障害発生状態で下記の操作を行った場合、モニタリソースの擬似障害が解除されます。

- WebManager(検証モード) で、モニタリソースの「擬似障害解除」を実行した場合
- WebManager のモードを、検証モード から 他のモードに変更する際に出力されるダイアログで「はい」を選択した場合
- clpmonctrl コマンドに –n オプション を指定した場合
- クラスタを停止した場合
- クラスタをサスペンドした場合

モニタリソースの監視プライオリティ

OS 高負荷時にモニタリソースへの監視を優先的に行うため、nice 値を設定することができます。

nice 値は 19(優先度低) ~ -20(優先度高) の範囲で指定することが可能です。

- nice値の優先度を上げることで監視タイムアウトの検出を抑制することが可能です。

モニタリソースの名前を変更するには

1. Builder の左部分に表示されているツリービューで [Monitors] のアイコンをクリックします。右部分のテーブルビューで、名前を変更したいモニタリソースのアイコンを右クリックし、[モニタリソースの名称変更] をクリックします。
2. [モニタリソース名の変更] ダイアログ ボックスが表示されます。変更する名前を入力します。

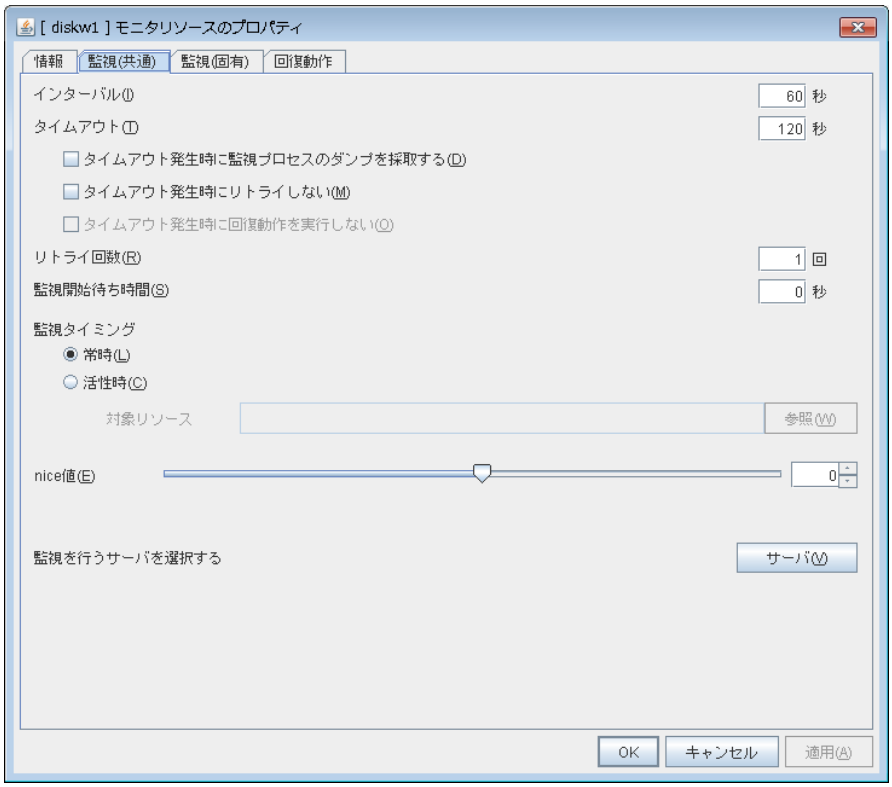
モニタリソースのコメントを表示 / 変更するには(モニタリソースのプロパティ)

1. Builder の左部分に表示されているツリービューで [Monitors] のアイコンをクリックします。右部分のテーブルビューで、コメントを変更したいモニタリソースのアイコンを右クリックし、[プロパティ] をクリックします。[モニタリソースのプロパティ] ダイアログボックスが表示されます。
2. [情報] タブに、モニタリソースの名前、コメントが表示されます。コメント **(127 バイト以内)** を入力 / 変更します。半角英数字のみ入力可能です。

注: [情報] タブではモニタリソース名の変更はできません。変更する場合は、上記ステップ 1 と同様に [Monitors] のアイコンを右クリックし、[モニタリソースの名称変更] をクリックして値を入力します。

モニタリソースの監視設定を表示 / 変更するには (モニタリソース共通)

1. Builder の左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のモニタリソースを右クリックし、[プロパティ] の [監視] タブをクリックします。
3. [監視] タブで、以下の説明に従い監視設定の表示 / 変更を行います。



インターバル(1～999)

監視対象の状態を確認する間隔を設定します。

タイムアウト(5～999 [1])

ここで指定した時間内に監視対象の正常状態が検出できない場合に異常と判断します。

タイムアウト発生時に監視プロセスのダンプを採取する

本機能を有効にした場合、モニタリソースがタイムアウトすると、タイムアウトしたモニタリソースのダンプが採取されます。ダンプ情報は最大 5 回採取されます。

[1] ユーザ空間モニタリソースで監視方法にipmiを設定している場合は、255以下の値を設定する必要があります。

タイムアウト発生時にリトライしない

本機能を有効にした場合、モニタリソースがタイムアウトすると即座に回復動作を実行します。

タイムアウト発生時に回復動作を実行しない

本機能を有効にした場合、モニタリソースがタイムアウトした場合に回復動作を実行しません。

また、タイムアウトが発生した場合にはリトライ回数の回数カウンタはリセットされます。

本機能は、[タイムアウト発生時にリトライしない] 機能を有効にしている場合のみ設定可能です。

注:

下記のモニタリソースでは、[タイムアウト発生時にリトライしない], [タイムアウト発生時に回復動作を実行しない] 機能は設定できません。

- ユーザ空間モニタリソース
- カスタムモニタリソース (監視タイプが [非同期] の場合のみ)
- マルチターゲットモニタリソース
- 仮想マシンモニタリソース
- 外部連携モニタリソース
- JVM モニタリソース
- システムモニタリソース

リトライ回数(0～999)

異常状態を検出後、連続してここで指定した回数の異常を検出したときに異常と判断します。

0 を指定すると最初の異常検出で異常と判断します。

監視開始待ち時間(0～9999)

監視を開始するまでの待ち時間を設定します。

注:

下記のモニタリソースでは、監視を開始するまでの待ち時間として、モニタタイムアウト時間 と 監視開始待ち時間の値が大きいほうを使用します。

- 外部連携モニタリソース
- カスタムモニタリソース (監視タイプが [非同期] の場合のみ)
- DB2 モニタリソース
- システムモニタリソース
- JVM モニタリソース
- MySQL モニタリソース
- Oracle モニタリソース
- PostgreSQL モニタリソース

- プロセス名モニタリソース
- Sybase モニタリソース

監視タイミング

監視のタイミングを設定します。

◆ [常時]
監視を常時行います。

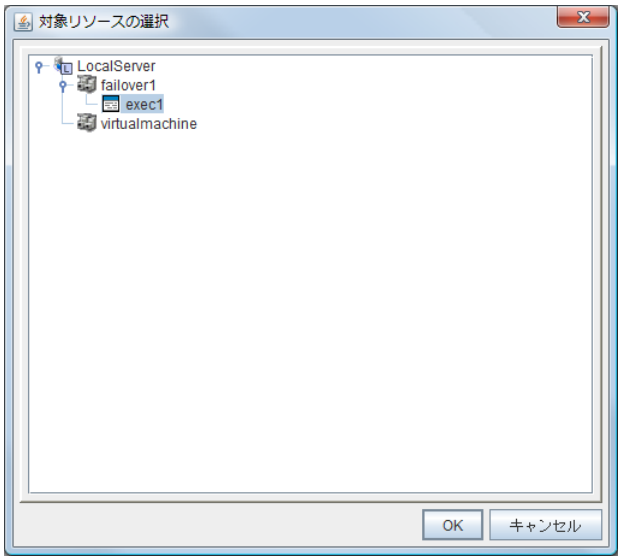
◆ [活性時]
指定したリソースが活性するまで監視を行いません。

対象リソース

活性時監視を行う場合に対象となるリソースを表示します。

参照

対象リソースの選択ダイアログ ボックスを表示します。LocalServer とクラスタに登録されているグループ名、リソース名がツリー表示されます。対象リソースとして設定するリソースを選択して[OK]をクリックします。



nice 値

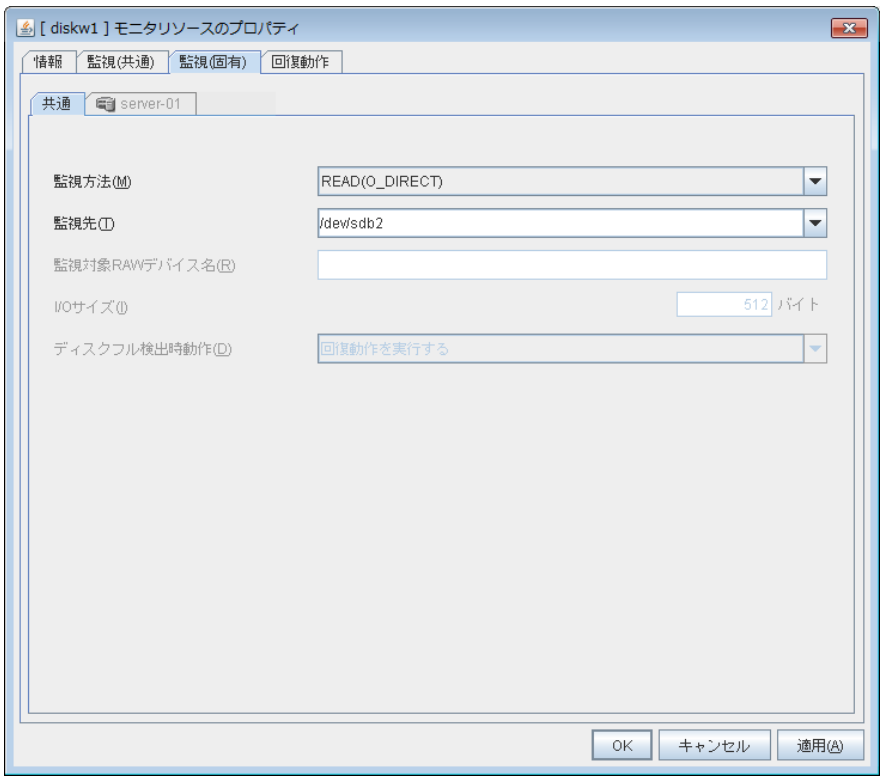
プロセスの nice 値を設定します。

ディスクモニタリソースの設定

ディスクモニタリソースは、ディスクデバイスの監視を行います。

ディスクモニタリソース (TUR 方式) が使用できないディスクでは、READ(O_DIRECT) 方式での監視を推奨します。

1. Builder の左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のディスクモニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



監視方法

ディスクデバイスを監視するときの監視方法を下記より選択します。

- TUR
- TUR(generic)
- TUR(legacy)
- READ
- READ (O_DIRECT)
- WRITE (FILE)
- READ (RAW)

- READ (VXVM)

監視先 (1023 バイト以内)

- 監視方法が WRITE (FILE)の場合

  監視用のファイルのパス名を指定します。[/] で始まる必要があります。

  ファイル名は絶対パスで指定してください。既存のファイルのファイル名を指定した場合には上書きされファイルの内容は失われます。

- 監視方法が READ(O_DIRECT)の場合

  監視用のデバイスファイルもしくはファイルのパス名を指定します。[/] で始まる必要があります。

  デバイスファイル名もしくはファイル名は絶対パスで指定してください。

  ファイル名を指定する場合は、指定するファイルを事前に作成しておく必要があります。

- 監視方法が READ(RAW)の場合

  監視先は空欄でもかまいません。ただし、監視対象RAWデバイス名の入力は必須となります。バインドをして監視を行う場合のみ指定します。既にmountしているパーティションデバイスまたはmountする可能性のあるパーティションデバイスをデバイス名に設定して監視することはできません。
  また、既にmountしているパーティションデバイスまたはmountする可能性のあるパーティションデバイスのwhole device(ディスク全体を示すデバイス)をデバイス名に設定して監視することもできません。監視専用のパーティションを用意してください。(監視用のパーティションサイズは、10MB以上を割り当ててください) [/] で始まる必要があります。

- 監視方法が READ(VXVM)の場合

  グレーアウトされて、選択できません。

- 監視方法が READ の場合

  ディスクデバイスを監視するときの監視先デバイス名もしくはファイル名を指定します。[/] で始まる必要があります。ファイル名を指定する場合は、指定するファイルを事前に作成しておく必要があります。

- 監視方法が上記以外の場合

  ディスクデバイスを監視するときの監視先デバイス名を指定します。[/] で始まる必要があります。

監視対象 RAW デバイス名 (1023 バイト以内)

監視方法に READ（RAW）、READ (VXVM)を選択した場合のみ入力可能となります。

- 監視方法がREAD(RAW)の場合

  raw アクセスするためのデバイス名を入力します。既にサーバプロパティの[ディスク I/F 一覧]に登録されている RAW デバイスは登録できません。VxVM ボリューム RAW デバイスの監視は監視方法に READ(VXVM)を選択してください。

- 監視方法がREAD(VXVM)の場合

  VxVM ボリューム RAW デバイス名を設定してください。ボリューム RAW デバイスのファイルシステムが vxfs ではない場合監視できません。[/] で始まる必要があります。

I/O サイズ(1～99999999)

監視処理で行う read または read/write のサイズを指定します。

＊READ(RAW), READ(VXVM), READ(O_DIRECT) を指定した場合、I/O サイズの入力項目はグレーアウトされます
対象のデバイスから 1 セクタ分の read を行います。
＊TUR, TUR (generic), TUR (legacy) を指定した場合、本設定項目は無視されます。

ディスクフル検出時動作

ディスクフル(監視するディスクに空き容量がない状態)検出時の動作を下記より選択します。

◆ 回復動作を実行する

ディスクモニタリソースはディスクフル検出時に異常として扱います。

◆ 回復動作を実行しない

ディスクモニタリソースはディスクフル検出時に警告として扱います。

＊READ, READ (RAW), READ (VXVM), READ (O_DIRECT), TUR, TUR (generic), TUR (legacy) を指定した場合、ディスクフル検出時動作の項目はグレーアウトされます

監視デバイス名にローカルディスクを設定すると、サーバのローカルディスク監視を行うことができます。

◆ ローカルディスク[/dev/sdb] を[READ 方式] で監視し、異常検出時に[OS 再起動] を行う設定例

設定項目	設定値	備考
監視デバイス名	/dev/sdb	2台目のSCSIディスク
監視方法	READ	READ 方式
回復対象	サーバ	ー
最終動作	サービス停止とOS再起動	OS再起動

ローカルディスク［/dev/sdb］を［TUR(generic) 方式］で監視し、異常検出時に［何もしない］(WebManagerへアラートの表示のみを行う)場合の設定例

設定項目	設定値	備考
監視デバイス名	/dev/sdb	2台目のSCSIディスク
監視方法	TUR(generic)	SG_IO方式
最終動作	何もしない	

ディスクモニタリソースによる監視方法

ディスクモニタリソースの監視方法は大きく分けて TUR と READ があります。

◆ TURの注意事項

- SCSI の Test Unit Ready コマンドや SG_IO コマンドをサポートしていないディスク、ディスクインタフェース(HBA) では使用できません。
ハードウェアがサポートしている場合でも、ドライバがサポートしていない場合があるので、ドライバの仕様も合わせて確認してください。
- LVM 論理ボリューム (LV) のデバイスでは ioctl が正常に実行できない可能性があるため、LV の監視には READ を使用してください。
- IDE インターフェイスのディスクの場合には、すべての TUR 方式は使用できません。

- S-ATA インターフェイスのディスクの場合には、ディスクコントローラのタイプや使用するディストリビューションにより、OS に IDE インターフェイスのディスク(hd)として認識される場合と SCSI インターフェイスのディスク(sd)として認識される場合があります。IDE インターフェイスとして認識される場合には、すべての TUR 方式は使用できません。SCSI インターフェイスとして認識される場合には、TUR(legacy)が使用できます。TUR(generic)は使用できません。
- Read 方式に比べて OS やディスクへの負荷は小さくなります。
- Test Unit Ready では、実際のメディアへの I/O エラーは検出できない場合があります。

- OS カーネルアップデート (kernel-2.6.18-274.18.1.el5以上、kernel-2.6.32-220.2.1.el6以上) を適用した環境ではディスク上のパーティションを監視対象に設定して使用することはできません。
- ディスク装置によっては TUR 発行時、装置の状態によって一時的に Unit Attention を返す場合があります。
  Unit Attention が一時的に返却されることは問題ではありませんが、TUR のリトライ回数を 0 回に設定している場合、上記をエラーと判断し、ディスクモニタリソースが異常となります。
  無用な異常検出を防ぐため、リトライ回数は 1 回以上を設定してください。

TUR の監視方法は、下記の 3 つが選択可能です。

◆ TUR

- 指定されたデバイスへ以下の手順で ioctl を発行して、その結果で判断します。ioctl(SG_GET_VERSION_NUM)コマンドを実行します。この ioctl の戻り値と SG ドライバの version を見て判断します。
- ioctl コマンド成功かつ SG ドライバの version が 3.0 以上なら SG ドライバを使用した ioctl TUR(SG_IO)を実行します。
- ioctl コマンド失敗または SG ドライバの version が 3.0 未満なら SCSI コマンドとして定義されている ioctl TUR を実行します。

◆ TUR(legacy)

- ioctl(Test Unit Ready)を使って監視を行います。指定されたデバイスへ SCSI コマンドとして定義されている Test Unit Ready(TUR)コマンドを発行してその結果で判断します。

◆ TUR(generic)

- ioctl TUR(SG_IO)を使って監視を行います。指定されたデバイスへ SCSI コマンドとして定義されている ioctl(SG_IO)コマンドを発行してその結果で判断します。SG_IO は SCSI ディスクであっても OS やディストリビューションによって動作しないことがあります。

READ の監視方法は、下記のとおりです。

◆ READ

- 指定されたデバイス(ディスクデバイスまたはパーティションデバイス) もしくはファイル上の指定されたサイズを read してその結果(read できたサイズ)で判断します。
- 指定されたサイズが read できたことを判断します。read したデータの正当性は判断しません。

- read するサイズを大きくすると OS やディスクへの負荷が大きくなります。
- read するサイズについては 99 ページの「ディスクモニタリソースで READ を選択した場合の I/O サイズ」を留意して設定してください。

READ(O_DIRECT) の監視方法は、下記のとおりです。

◆ READ (O_DIRECT)

- 指定されたデバイス(ディスクデバイスまたはパーティションデバイス) 上の 1 セクタ分もしくはファイルを、キャッシュを使用しない(O_DIRECT モード)で read してその結果(read できたサイズ)で判断します。
- read できたことを判断します。read したデータの正当性は判断しません。

READ (RAW) の監視方法は、下記のとおりです。

◆ READ (RAW)

- 監視方法「READ(O_DIRECT)」と同様に OS のキャッシュを使用しないで指定されたデバイスの read の監視を行います。
- read できたことを判断します。read したデータの正当性は判断しません。
- 監視方法「READ(RAW)」を設定する場合、既に mount しているパーティションまたは mount する可能性のあるパーティションの監視はできません。また、既に mount しているパーティションまたは mount する可能性のあるパーティションの whole device(ディスク全体を示すデバイス)を監視することもできません。監視専用のパーティションを用意してディスクモニタリソースに設定してください。(監視用のパーティションサイズは、10MB 以上を割り当ててください)

READ (VXVM) の監視方法は、下記のとおりです。

◆ READ (VXVM)

- 監視方法「READ(O_DIRECT)」と同様に OS のキャッシュを使用しないで指定されたデバイスの read の監視を行います。
- read できたことを判断します。read したデータの正当性は判断しません。
- ボリューム RAW デバイスのファイルシステムが vxfs ではない場合、「READ (VXVM)」で監視できません。

WRITE (FILE) の監視方法は、下記のとおりです。

◆ WRITE (FILE)

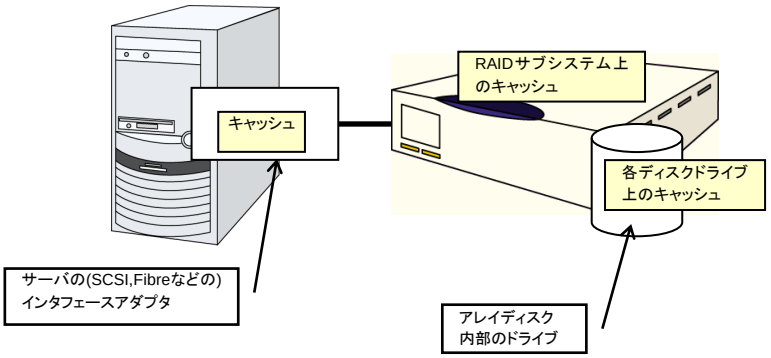
- 指定されたパス名のファイルを作成、書き込み、削除を行い判断します。
- 書き込んだデータの正当性は判断しません。

ディスクモニタリソースで READ を選択した場合の I/Oサイズ

監視方法で READ を選択した場合の read を行うサイズを指定します。

使用するディスクやインターフェイスにより、様々な read 用のキャッシュが実装されている場合があります。そのため I/O サイズが小さい場合にはキャッシュにヒットしてしまい read のエラーを検出できない場合があります。

READ の I/O のサイズはディスクの障害を発生させて障害の検出ができることを確認して指定してください。

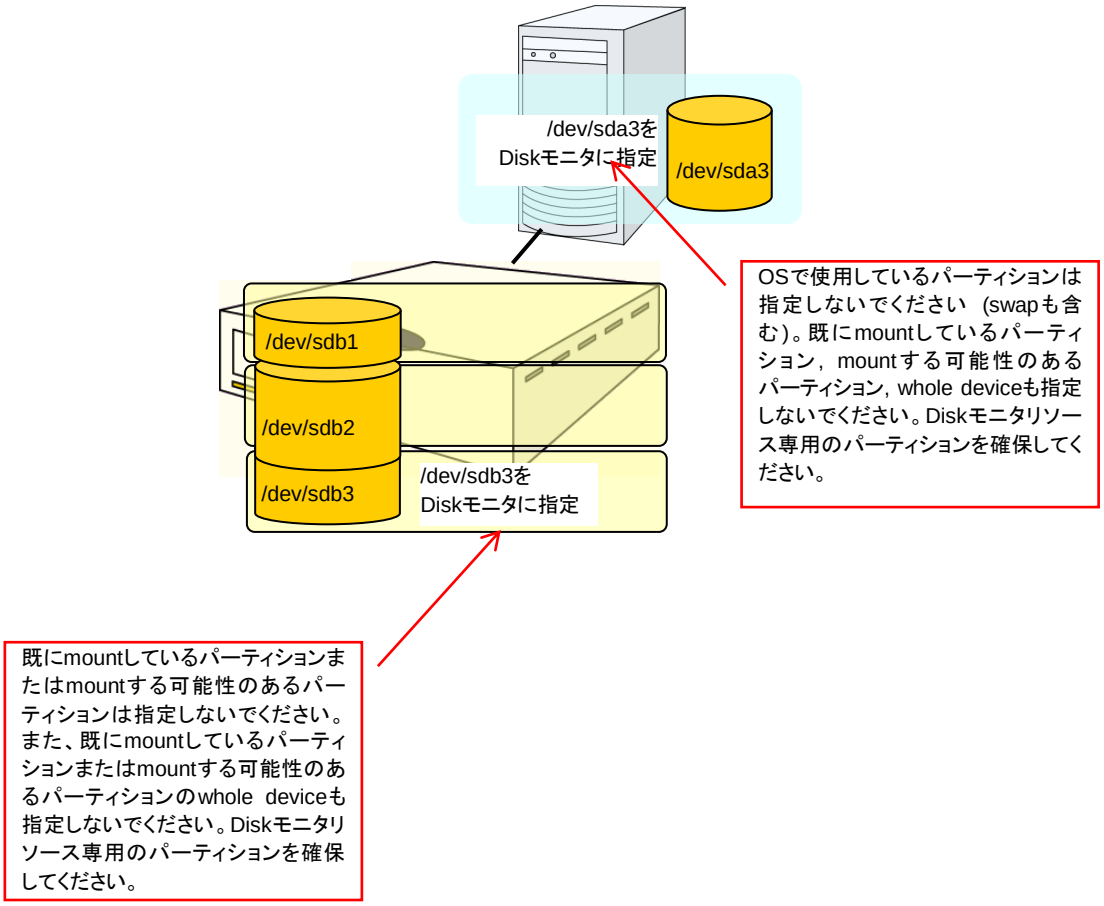


(注意) 上の図は共有ディスクの一般的な概念図を表したもので、必ずしもすべてのアレイ装置に当てはまるものではありません

ディスクモニタリソースで READ(RAW) を選択した場合の設定例

ディスクモニタの設定例

- Diskモニタリソース (内蔵HDDを「READ(RAW)」で監視)
- Diskモニタリソース (共有ディスクを「READ(RAW)」で監視)

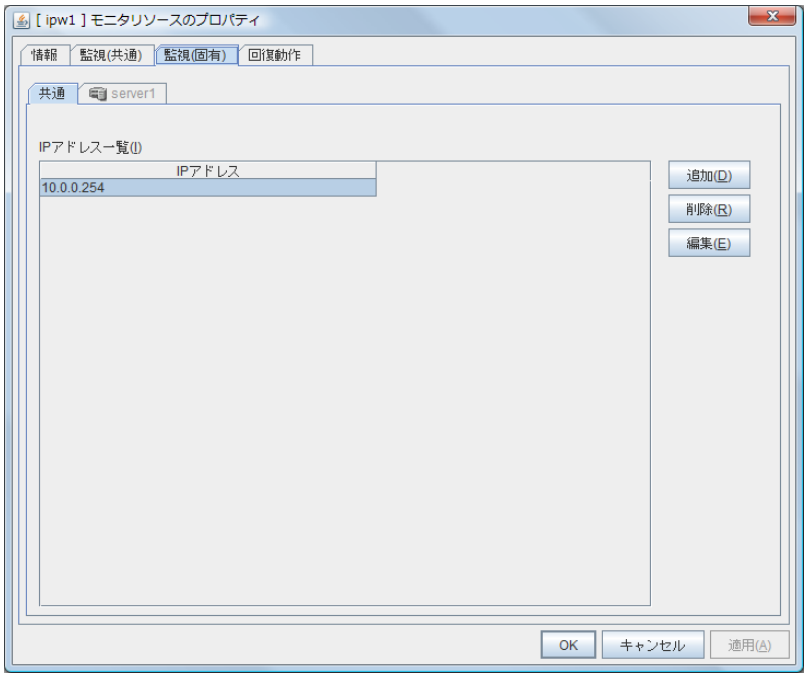


IP モニタリソースの設定

IP モニタリソースとは、ping コマンドを使用して、IP アドレスの監視を行うモニタリソースです。

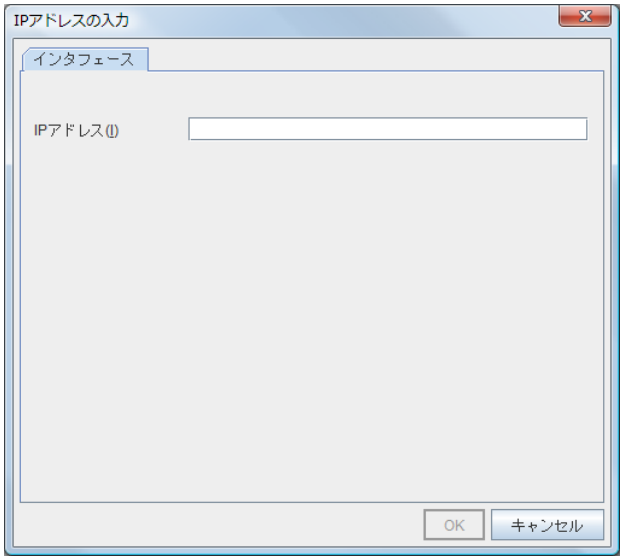
1. Builder の左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の IP モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。

[IP アドレス一覧] には監視する IP アドレスの一覧が表示されます。



追加

監視する IP アドレスを追加します。IP アドレスの入力ダイアログ ボックスが表示されます。



IP アドレス(255 バイト以内)

監視を行う IP アドレスまたはホスト名を入力して[OK]を選択してください。パブリック LAN に存在する実 IP アドレスまたはホスト名を入力してください。ホスト名を設定する場合は、OS 側に名前解決の設定(/etc/hosts へのエントリの追加など)をしてください。

削除

[IP アドレス一覧] で選択している IP アドレスを監視対象から削除します。

編集

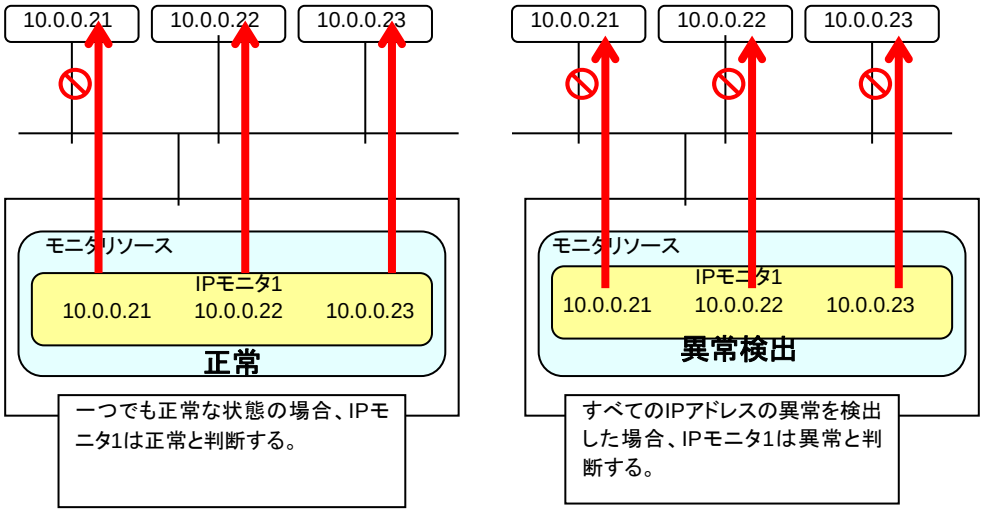
IP アドレスの入力ダイアログ ボックスが表示されます。[IP アドレス一覧] で選択している IP アドレスが表示されるので、編集して[OK]を選択します。

IP モニタリソースの監視方法

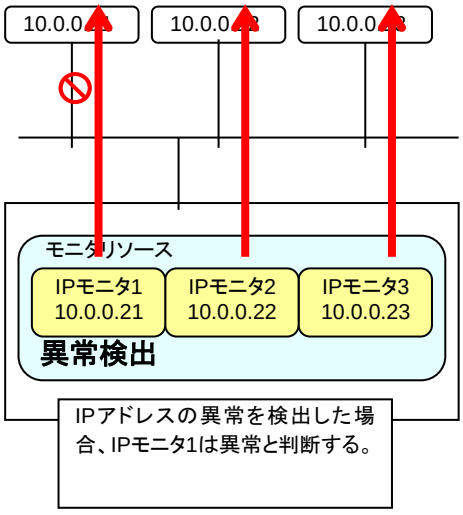
指定した IP アドレスを ping コマンドで監視します。指定した IP アドレスすべての応答がない場合に異常と判断します。

IP アドレスの応答確認には ICMP の packet type 0 (Echo Reply) と 8 (Echo Request) が使用されます。

◆ 複数のIPアドレスについてすべてのIPアドレスが異常時に異常と判断したい場合、1 つのIPモニタリソースにすべてのIPアドレスを登録してください。

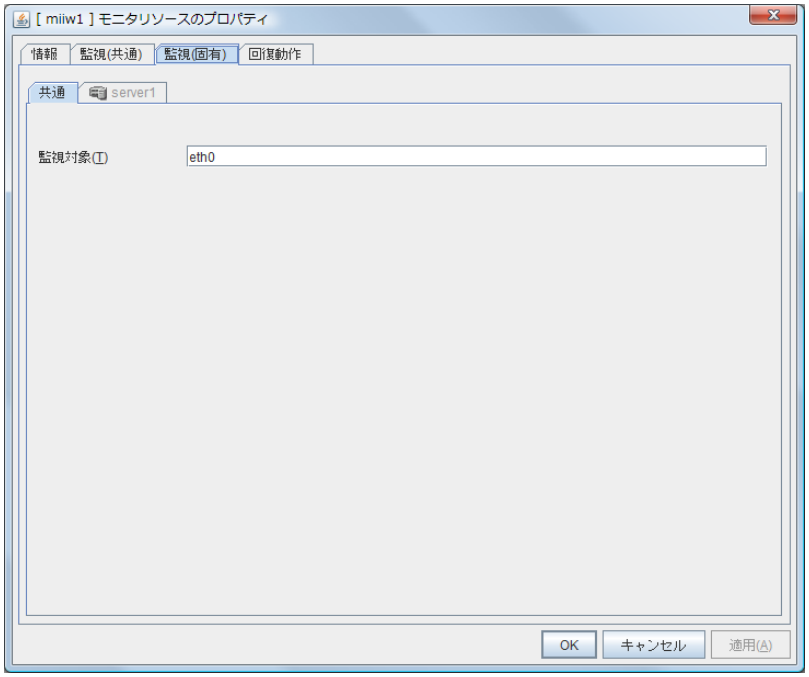


◆ 複数のIPアドレスについてどれか1つが異常時に異常と判断したい場合、個々のIPアドレスについて1つずつのIPモニタリソースを作成してください。



NIC Link Up/Down モニタリソースの設定

NIC Link Up/Down　モニタリソースは、指定したNICのLink状態を取得し、Linkの Up/Downを監視します。



監視対象(15 バイト以内)

監視を行う NIC のインターフェイス名を設定してください。ボンドデバイス(例:bond.600)、およびチームデバイス(例:team0)共に監視可能です。VLAN, tagVLAN の監視も可能です(設定例:eth0.8)。

NIC Link UP/Down モニタリソースの動作環境

NIC Link UP/Down モニタリソースをサポートするネットワークインターフェイス

NIC Link UP/Down モニタリソースは、以下のネットワークインターフェイスで動作確認しています。

Ethernet Controller(Chip)	Bus	Driver version
Intel 82557/8/9	PCI	3.5.10-k2-NAPI
Intel 82546EB	PCI	7.2.9
Intel 82546GB	PCI	7.3.20-k2-NAPI 7.2.9
Intel 82573L	PCI	7.3.20-k2-NAPI
Intel 80003ES2LAN	PCI	7.3.20-k2-NAPI
Broadcom BCM5721	PCI	7.3.20-k2-NAPI

NIC Link UP/Down モニタリソースの注意事項

NIC のボード、ドライバによっては、必要な ioctl( )がサポートされていない場合があります。
NIC Link Up/Down モニタリソースの動作可否は、各ディストリビュータが提供する ethtool コマンドで確認することができます。

```
ethtool eth0
Settings for eth0:
     Supported ports: [ TP ]
     Supported link modes:   10baseT/Half 10baseT/Full
                             100baseT/Half 100baseT/Full
                             1000baseT/Full
     Supports auto-negotiation: Yes
     Advertised link modes:  10baseT/Half 10baseT/Full
                             100baseT/Half 100baseT/Full
                             1000baseT/Full
     Advertised auto-negotiation: Yes
     Speed: 1000Mb/s
     Duplex: Full
     Port: Twisted Pair
     PHYAD: 0
     Transceiver: internal
     Auto-negotiation: on
     Supports Wake-on: umbg
     Wake-on: g
     Current message level: 0x00000007 (7)
     Link detected: yes
```

- ethtoolコマンドの結果で LANケーブルのリンク状況 ("Link detected: yes") が表示されない場合
  - CLUSTERPRO の NIC Link Up/Down モニタリソースが動作不可能な可能性が高いです。IP モニタリソースで代替してください。

- ethtoolコマンドの結果で LANケーブルのリンク状況 ("Link detected: yes") が表示される場合
  - 多くの場合 CLUSTERPRO の NIC Link Up/Down モニタリソースが動作可能ですが、希に動作不可能な場合があります。
  - 特に以下のようなハードウェアでは動作不可能な場合があります。IP モニタリソースで代替してください。
  - ブレードサーバのように実際の LAN のコネクタと NIC のチップとの間にハードウェアが実装されている場合
  - 監視対象の NIC が Bonding 環境の場合、MII Polling Interval の設定値が 0 以上に設定されているか確認してください。

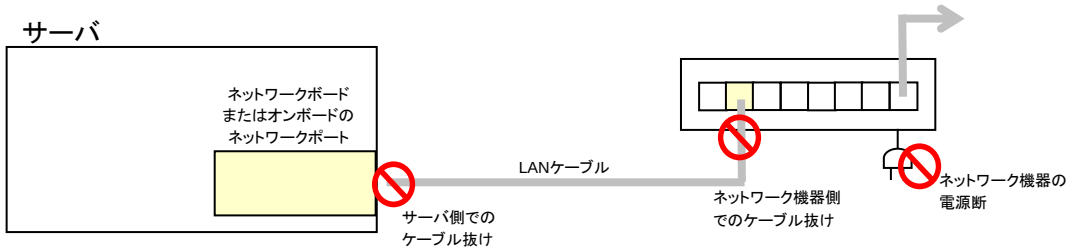
実機で CLUSTERPRO を使用して NIC Link Up/Down モニタリソースの使用可否を確認する場合には以下の手順で動作確認を行ってください。

1. NIC Link Up/Down モニタリソースを構成情報に登録してください。
   NIC Link Up/Down モニタリソースの異常検出時回復動作の設定は「何もしない」を選択してください。
2. サーバを起動してください。

3. NIC Link Up/Down モニタリソースのステータスを確認してください。
LAN ケーブルのリンク状態が正常状態時に NIC Link Up/Down モニタリソースのステータスが異常となった場合、NIC Link Up/Down モニタリソースは動作不可です。

4. LAN ケーブルのリンク状態を異常状態(リンクダウン状態)にしたときに NIC Link Up/Down モニタリソースのステータスが異常となった場合、NIC Link Up/Down モニタリソースは動作可能です。
ステータスが正常のまま変化しない場合、NIC Link Up/Down モニタリソースは動作不可です。

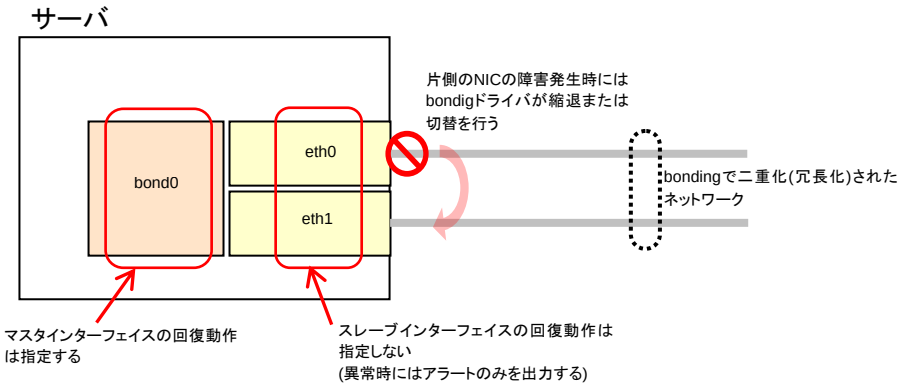
NIC Link UP/Down 監視の構成および範囲



- NIC のドライバへの ioctl( )によりネットワーク(ケーブル)のリンク確立状態を検出します。
(IP モニタの場合は、指定された IP アドレスへの ping の反応で判断をします。)
- インタコネクト(ミラーコネクト)専用の NIC を監視することもできますが、2 ノード間をクロスケーブルで直結している場合には 片サーバダウン時に(リンクが確立しないため)残りのサーバ側でも 異常を検出します。
監視異常時の回復動作の設定は適切な値を設定するように注意してください。
たとえば、最終動作に"クラスタサービス停止と OS 再起動"すると、残りのサーバ側は無限に OS 再起動を繰り返すことになります。

また、ネットワークを bonding 化している場合には、bonding による可用性を活かしたまま下位のスレーブインターフェイス(eth0, eth1...)だけでなくマスタインターフェイス(bond0...)も監視することが可能です。その場合には、下記の設定を推奨します。

- スレーブインターフェイス
  - 異常検出時の回復動作:何もしない
    片方のネットワークケーブルのみ(eth0)の異常時には CLUSTERPRO は回復動作を実行せず、アラートのみ出力します。
    ネットワークの回復動作は、bonding が行います。
- マスタインターフェイス
  - 異常検出時の回復動作:シャットダウンなどを設定する
    全てのスレーブインターフェイスの異常時(マスタインターフェイスがダウン状態)に CLUSTERPRO は、回復動作を実行します。


サーバ
bond0
eth0
eth1
片側のNICの障害発生時には
bondigドライバが縮退または
切替を行う
bondingで二重化(冗長化)された
ネットワーク
マスタインターフェイスの回復動作
は指定する
スレーブインターフェイスの回復動作は
指定しない
(異常時にはアラートのみを出力する)

PID モニタリソースの設定

活性に成功した EXEC リソースを監視します。プロセス ID の有無を監視することによってプロセス ID の消滅時に異常と判断します。

監視を行う EXEC リソースは 264 ページの「モニタリソース共通の設定」の [対象リソース] で設定します。EXEC リソースの起動時の設定が[非同期]の場合のみ監視できます。プロセスのストールを検出することは出来ません。

注: データベース、samba、apache、sendmail などのストール監視を行うには「CLUSTERPRO 監視オプション」を購入してください。

PIDモニタリソースの注意事項

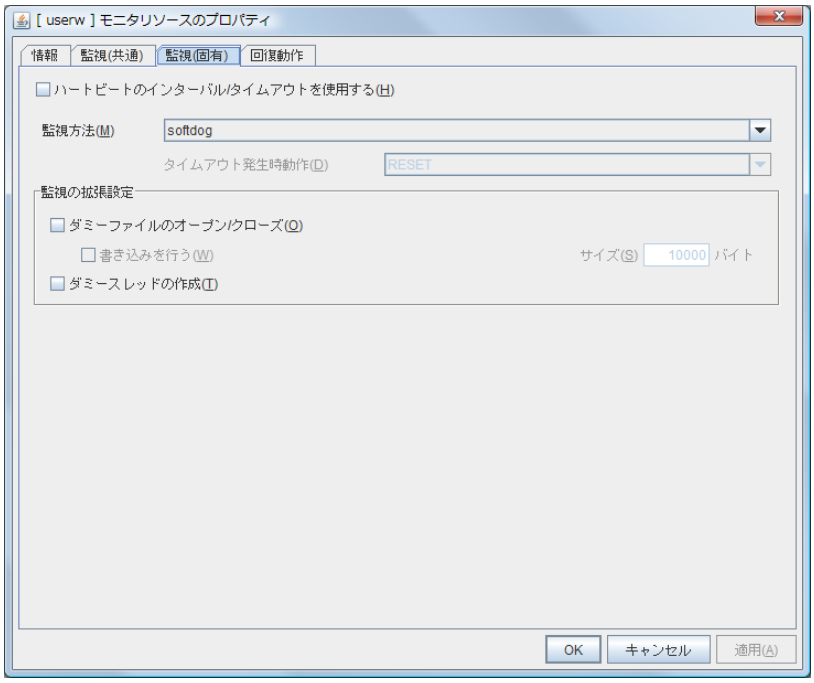
活性に成功した EXEC リソースを監視します。EXEC リソースの開始スクリプトの起動時の設定が[非同期]の場合のみ監視できます。

ユーザ空間モニタリソースの設定

ユーザ空間モニタリソースは、ユーザ空間のストールを異常として判断します。

本リソースは自動的に登録されます。監視方法は softdog のユーザ空間モニタリソースが自動登録されます。

1. Builder の左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のユーザ空間モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



ハートビートのインターバル/タイムアウトを使用する

監視のインターバルとタイムアウトを、ハートビートのインターバルとタイムアウトを使用するかどうかを設定します。

- チェックボックスがオン

  ハートビートのインターバルとタイムアウトを使用します。

- チェックボックスがオフ

  ハートビートの設定は使用せず、監視タブで設定するインターバルとタイムアウトを使用します。タイムアウトはインターバルより大きい値を設定する必要があります。[監視方法] に ipmi を設定している場合、タイムアウトを 255 以下に設定する必要があります。

監視方法

ユーザ空間モニタリソースの監視方法を以下の中から選択します。既に他のユーザ空間モニタリソースで使用している監視方法は選択できません。

- softdog

  softdog ドライバを使用します。

- ipmi

  ipmiutil を使用します。

- keepalive

  clpkhb ドライバ、clpka ドライバを使用します。

- none

  何も使用しません。

タイムアウト発生時動作

最終動作を設定します。監視方法が keepalive の場合にのみ設定可能です。

- RESET

  サーバをリセットします。

- PANIC

  サーバをパニックさせます。

ダミーファイルのオープンクローズ

監視を行う際、インターバルごとにダミーファイルのオープン/クローズを行うかどうかを設定します。

- チェックボックスがオン

  ダミーファイルのオープン/クローズを行います。

- チェックボックスがオフ

  ダミーファイルのオープン/クローズを行いません。

書き込みを行う

ダミーファイルのオープン/クローズを行う場合に、ダミーファイルに書き込みを行うかどうかを設定します。

- チェックボックスがオン

  ダミーファイルの書き込みを行います。

- チェックボックスがオフ

  ダミーファイルの書き込みを行いません。

サイズ(1～9999999)

ダミーファイルに書き込みを行う場合に書き込むサイズを設定します。

ダミースレッドの作成

監視を行う際にダミースレッドの作成を行うかどうかを設定します。

- チェックボックスがオン

  ダミースレッドの作成を行います。

- チェックボックスがオフ

  ダミースレッドの作成を行いません。

ユーザ空間モニタリソースが依存するドライバ

監視方式 softdog

softdog

- 監視方法がsoftdogの場合、このドライバが必要です。
- ローダブルモジュール構成にしてください。スタティックドライバでは動作しません。
- softdogドライバが使用できない場合、監視を開始することはできません。

監視方式 keepalive

clpka

clpkhb

- 監視方法がkeepaliveの場合、CLUSTERPROのclpkhbドライバ、clpkaドライバが必要です。
- clpkaドライバとclpkhbドライバはCLUSTERPROが提供するドライバです。サポート範囲については『インストールガイド』の『動作可能なディストリビューションとkernel』を参照してください。
- clpkhbドライバ、clpkaドライバが使用できない場合、監視を開始することはできません。

ユーザ空間モニタリソースが依存する rpm

監視方式 ipmi

ipmiutil

- 監視方法がipmiの場合、このrpmをインストールしておく必要があります。
- このrpmがインストールされていない場合、監視を開始することはできません。

ユーザ空間モニタリソースの監視方法

ユーザ空間モニタリソースの監視方法は以下のとおりです。

監視方法 softdog

監視方法が softdog の場合、OS の softdog ドライバを使用します。

監視方法 ipmi

監視方法が ipmi の場合、ipmiutil を使用します。

ipmiutil がインストールされていない場合、インストールする必要があります。

監視方法 keepalive

監視方法が keepalive の場合、clpkhb ドライバと clpka ドライバを使用します。

注: clpkhb ドライバ, clpka ドライバが動作するディストリビューション、kernel バージョンについては必ず『インストールガイド』の『動作可能なディストリビューションと kernel』で確認してください。
ディストリビュータがリリースするセキュリティパッチを既に運用中のサーバへ適用する場合(kernel バージョンが変わる場合)にも確認してください。

監視方法 none

監視方法 none は、評価用の設定です。ユーザ空間モニタリソースの拡張設定の動作だけを実行します。本番環境では設定しないでください。

ユーザ空間モニタリソースの拡張設定

ユーザ空間モニタリソースを拡張させる設定として、ダミーファイルのオープン/クローズ、ダミーファイルへの書き込み、ダミースレッドの作成があります。各設定に失敗するとタイマの更新を行いません。設定したタイムアウト値またはハートビートタイムアウト時間内に各設定が失敗し続けると OS をリセットします。

ダミーファイルのオープン/クローズ

監視間隔ごとにダミーファイルの作成、ダミーファイルの open、ダミーファイルの close、ダミーファイルの削除を繰り返します。

- この拡張機能を設定している場合、ディスクの空き容量がなくなるとダミーファイルの openに失敗してタイマの更新が行われず、OSをリセットします。

ダミーファイルへの書き込み

監視間隔毎にダミーファイルに設定したサイズを書き込みます。

- この拡張機能は、ダミーファイルのオープン/クローズが設定されていないと設定できません。

ダミースレッドの作成

監視間隔ごとにダミースレッドを作成します。

ユーザ空間モニタリソースのロジック

監視方法の違いによる処理内容、特徴は以下の通りです。シャットダウン監視では各処理概要のうち 1 のみの挙動になります。

監視方法 ipmi

- 処理概要

  以下の 2～7 の処理を繰り返します。

  1. IPMI タイマセット
  2. ダミーファイルの open()
  3. ダミーファイル write()
  4. ダミーファイル fdatasync()
  5. ダミーファイルの close()
  6. ダミースレッド作成
  7. IPMI タイマ更新

  - 処理概要 2～6 は監視の拡張設定の処理です。各設定を行っていないと処理を行いません。

- タイムアウトしない(上記2～7が問題なく処理される)場合の挙動
  リセットなどのリカバリ処理は実行されません
- タイムアウトした(上記2～7のいずれかが停止または遅延した)場合の挙動
  BMC(サーバ本体のマネージメント機能)によりリセットを発生させます
- メリット
  - BMC(サーバ本体のマネージメント機能)を使用するため kernel 空間の障害を受けにくく、リセットができる確率が高くなります。
- デメリット
  - H/W に依存しているため IPMI をサポートしていないサーバ、ipmiutil が動作しないサーバでは使用できません。
  - ESMPRO/ServerAgent を使用しているサーバでは使用できません。
  - その他サーバベンダが提供するサーバ監視ソフトウェアと共存できない可能性があります。
  - 一部のアーキテクチャでは ipmiutil が提供されていません。

監視方法 softdog

◆ 処理概要

以下の 2～7 の処理を繰り返します。

1. softdog セット
2. ダミーファイルの open()
3. ダミーファイル write()
4. ダミーファイル fdatasync()
5. ダミーファイルの close()
6. ダミースレッド作成
7. softdog タイマ更新

・ 処理概要 2～6 は監視の拡張設定の処理です。各設定を行っていないと処理を行いません。

◆ タイムアウトしない(上記 2～7 が問題なく処理される)場合の挙動
リセットなどのリカバリ処理は実行されません

◆ タイムアウトした(上記 2～7 のいずれかが停止または遅延した)場合の挙動
softdog.ko によりリセットを発生させます

◆ メリット

・ H/W に依存しないため softdog kernel モジュールがあれば使用できます。
(一部のディストリビューションではデフォルトで softdog が用意されていないものもがありますので 設定する前に softdog の有無を確認してください)

◆ デメリット

・ softdog が kernel 空間のタイマロジックに依存しているため kernel 空間に障害が発生した場合にリセットされない場合があります。

監視方法 keepalive

◆ 処理概要

以下の 2～7 の処理を繰り返します。

1. keepalive タイマセット
2. ダミーファイルの open()
3. ダミーファイル write()
4. ダミーファイル fdatasync()
5. ダミーファイルの close()
6. ダミースレッド作成
7. keepalive タイマ更新

・ 処理概要 2～6 は監視の拡張設定の処理です。各設定を行っていないと処理を行いません。

- タイムアウトしない(上記 2～7 が問題なく処理される)場合の挙動
  リセットなどのリカバリ処理は実行されません
- タイムアウトした(上記 2～7 のいずれかが停止または遅延した)場合の挙動
  - アクションの設定にしたがって、clpka.ko によりリセットまたはパニックを発生させます
- メリット
  - アクションとしてパニックが設定できます。
- デメリット
  - 動作できる(ドライバを提供している)ディストリビューション、アーキテクチャ、カーネルバージョンが制限されます。
  - clpka が kernel 空間のタイマロジックに依存しているため kernel 空間に障害が発生した場合にリセットされない場合があります。

ipmi動作可否の確認方法

サーバ本体の ipmiutil の対応状況を確認する方法は、以下の手順で簡易確認することができます。

1. ダウンロードした ipmiutil の rpm パッケージをインストールする。[2]
2. /usr/sbin/wdt または/usr/sbin/iwdt を実行する。
3. 実行結果を確認する。

以下のように表示される場合(/usr/sbin/wdt の実行結果)
(以下は表示例です。H/W により表示される値が異なる場合があります。)

```
wdt ver 1.8
-- BMC version 0.8, IPMI version 1.5
wdt data: 01 01 01 00 31 17 31 17
Watchdog timer is stopped for use with BIOS FRB2. Logging
            pretimeout is 1 seconds, pre-action is None
            timeout is 593 seconds, counter is 593 seconds
            action is Hard Reset
```

ipmiutil は使用できます。監視方法に ipmi を選択することが可能です。

以下のように表示される場合(/usr/sbin/wdt の実行結果)

```
wdt version 1.8
ipmignu_cmd timeout, after session activated
```

ipmiutil は使用できません。監視方法に ipmi を選択しないでください。

[2]一部のディストリビューションではディストリビューションと共にインストールされています。その場合には ipmi-util rpmパッケージのインストールは不要です。

使用しているipmiコマンド

ユーザ空間モニタリソース、シャットダウン監視では ipmiutil のうち以下のコマンド、オプションを使用します。

コマンド	オプション	使用するタイミング	
		ユーザモードストール監視	シャットダウン監視
wdt iwdt	-e (タイマ開始)	開始時	監視開始時
	-d (タイマ停止)	終了時	終了時(SIGTERM有効)
	-r (タイマ更新)	開始時/監視間隔毎	監視開始時
	-t (タイムアウト値セット)	開始時/監視間隔変更時	監視開始時

ユーザ空間モニタリソースの注意事項

全監視方法での共通の注意事項

- Builder で設定情報を作成すると監視方法 softdog のユーザ空間モニタリソースが自動で作成されます。
- 監視方法の異なるユーザ空間モニタリソースを追加することができます。自動的に作成された監視方法 softdog のユーザ空間モニタリソースは削除することもできます。
- OS の softdog ドライバが存在しない、または CLUSTERPRO の clpkhb ドライバ、clpka ドライバが存在しない、ipmiutil の rpm がインストールされていないなどの理由によりユーザ空間モニタリソースの活性に失敗した場合、WebManager のアラートビューに”Monitor userw failed.” というメッセージが表示されます。WebManager のツリービュー、clpstat コマンドでの表示ではリソースステータスは[正常] が表示され、サーバのステータスは[停止済] が表示されます。

ipmi による監視の注意事項

- ipmi に関する注意事項は「モニタリソースの異常検出時の設定を表示 / 変更するには(モニタリソース共通)」の「使用している ipmi コマンド」を参照してください。

以下の組み合わせにて動作検証を行いました。

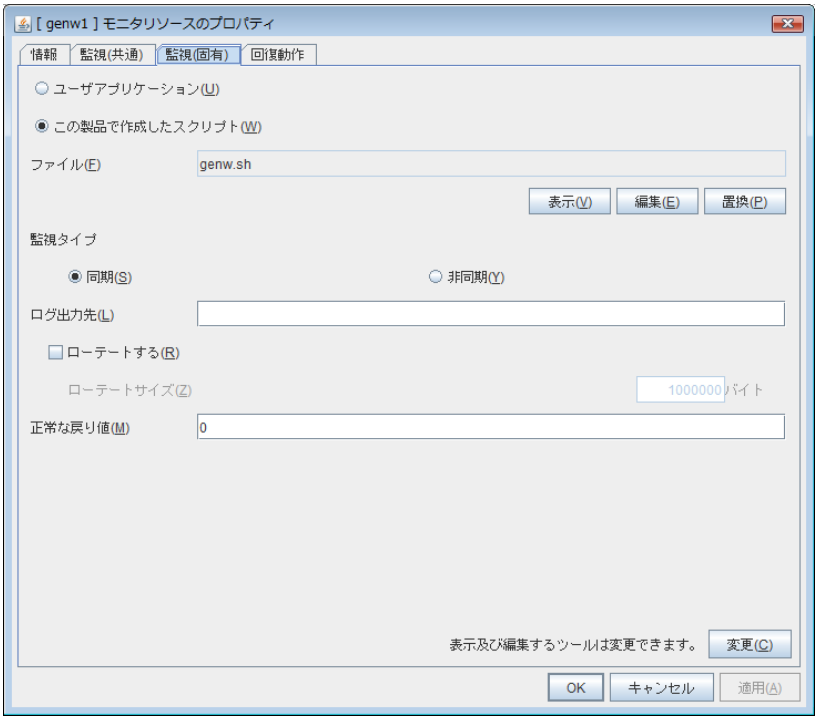
ディストリビューション	kernel version	ipmiutilのバージョン	サーバ
Red Hat Enterprise Linux AS 5 (update1)	2.6.18-53.el5	ipmiutil-1.7.9-1.x86_64.rpm	Express5800/120Rg-1
Red Hat Enterprise Linux AS 4 (update6)	2.6.9-67.EL smp	ipmiutil-2.0.8-1.x86_64.rpm	Express5800/120Rg-1
Asianux Server 3	2.6.18-8.10AXxen	ipmiutil-1.7.9-1.x86_64.rpm	Express5800/120Rg-2
Red Hat Enterprise Linux AS 5(update4)	2.6.18-164.el5	Ipmiutil-2.6.1-1.x86_64.rpm	Express5800/120Rf-1

注: ESMPRO/ServerAgent などサーバベンダが提供するサーバ監視ソフトウェアを使用する場合には、監視方法に IPMI を選択しないでください。
これらのサーバ監視ソフトウェアと ipmiutil は共にサーバ上の BMC(Baseboard Management Controller)をするため、競合が発生して正しく監視が行うことができなくなります。

カスタムモニタリソースの設定

カスタムモニタリソースは、任意のスクリプトを実行することによりシステムモニタを行うモニタリソースです。

1. Builder 左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のカスタム監視リソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います



ユーザアプリケーション

スクリプトとしてサーバ上の実行可能ファイル(実行可能なシェルスクリプトファイルや実行ファイル)を使用します。各実行可能ファイル名は、サーバ上のローカルディスクの絶対パスで設定します。

各実行可能ファイルは、Builder の構成情報には含まれません。Builder で編集やアップロードはできませんので、サーバ上に準備する必要があります。

この製品で作成したスクリプト

スクリプトとして Builder で準備したスクリプトファイルを使用します。必要に応じて Builder でスクリプトファイルを編集できます。スクリプトファイルは、構成情報に含まれます。

ファイル(1023 バイト以内)

[ユーザアプリケーション]を選択した場合に、実行するスクリプト（実行可能なシェルスクリプトファイルや実行ファイル）を、サーバ上のローカルディスクの絶対パスで設定します。

表示

[この製品で作成したスクリプト]を選択した場合に、スクリプトファイルをエディタで表示します。エディタで編集して保存した内容は反映されません。表示しようとしているスクリプトファイルが表示中または編集中の場合は表示できません。

編集

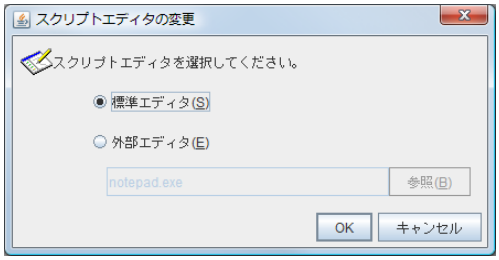
[この製品で作成したスクリプト]を選択した場合に、スクリプトファイルをエディタで編集します。変更を反映するには上書き保存を実行してください。編集しようとしているスクリプトファイルが表示中または編集中の場合は編集できません。スクリプトファイル名の変更はできません。

置換

[この製品で作成したスクリプト]を選択した場合に、スクリプトファイルの内容を、ファイル選択ダイアログ ボックスで選択したスクリプトファイルの内容に置換します。スクリプトが表示中または編集中の場合は置換できません。ここではスクリプトファイルを選択してください。バイナリファイル(アプリケーションなど)は選択しないでください。

変更

スクリプトエディタの変更ダイアログが表示されます。スクリプトを表示または編集するエディタを任意のエディタに変更できます。



標準エディタ

スクリプトエディタに標準のエディタを使用します。

- Linux … vi(実行ユーザのサーチパスで検索される vi)
- Windows … メモ帳(実行ユーザのサーチパスで検索される notepad.exe)

外部エディタ

スクリプトエディタを任意に指定します。[参照]を選択し、使用するエディタを指定します。

Linux でCUI ベースのエディタを外部エディタで指定するにはシェルスクリプトを作成してください。

以下の例は vi を実行するシェルスクリプトです。

xterm -name clpedit -title "Cluster Builder" -n "Cluster Builder" -e vi "$1"

監視タイプ

監視の方法を選択します。

- 同期(既定値)

  定期的にスクリプトを実行し、そのエラーコードにより異常の有無を判断します。

- 非同期

  監視開始時にスクリプトを実行し、そのプロセスが消失した場合に異常と判断します。

ログ出力先(1023 バイト以内)

スクリプト内で出力するログの出力先を設定します。

[ローテートする] のチェックボックスがオフの場合は無制限に出力されますのでファイルシステムの空き容量に注意してください。

[ローテートする] のチェックボックスがオンの場合は、出力されるログファイルは、ローテートします。また、以下の注意事項があります。

[ログの出力先] には 1009 バイト以内でログのパスを記述してください。1010 バイトを超えた場合、ログの出力が行えません。

ログファイルの名前の長さは 31 バイト以内で記述してください。32 バイト以上の場合、ログの出力が行えません。

複数のカスタムモニタリソースでログローテートを行う場合、パス名が異なってもログファイルの名前が同じ場合、(ex. /home/foo01/log/genw.log, /home/foo02/log/genw.log) ローテートサイズが正しく反映されないことがあります。

ローテートする

スクリプトや実行可能ファイルの実行ログを、オフの場合は無制限のファイルサイズで、オンの場合はローテートして出力します。

ローテートサイズ (1～999999999)

[ローテートする]チェックボックスがオンの場合に、ローテートするサイズを指定します。

ローテート出力されるログファイルの構成は、以下のとおりです。

ファイル名	内容
[ログ出力先]指定のファイル名	最新のログです。
[ログ出力先]指定のファイル名.pre	ローテートされた以前のログです。

正常な戻り値(1023 バイト以内)

監視タイプが[同期]の場合にスクリプトのエラーコードがどのような値の場合に正常と判断するかを設定します。複数の値がある場合は、0,2,3 というようにカンマで区切るか、 0-3 のようにハイフンで値の範囲を指定します。

既定値 : 0

カスタムモニタリソースの注意事項

監視タイプが [非同期] の場合、監視リトライ回数を 1 回以上に設定すると、正常に監視を行うことができません。監視タイプを [非同期] に設定する場合は、監視リトライ回数を 0 回に設定してください。

CLUSTERPRO SingleServerSafe X3.0.4-1 以前のバージョンでは、モニタリソースの監視設定の「タイムアウト発生時に監視プロセスのダンプを採取する」を設定することができましたが、カスタムモニタリソースでは有益な情報が得られる機能ではありませんでした。そのため、CLUSTERPRO SingleServerSafe X3.1.0-1 以降のバージョンでは、カスタムモニタリソースでの設定可能な機能から削除しました。
代替機能を設定したい場合は、カスタムモニタリソースの「ログ出力先」を設定してログ出力するようにしてください。

スクリプトのログローテート機能を有効にした場合、スクリプト終了時に指定されたファイルへの書き込みが行われます。監視タイプを [非同期] に設定している場合、スクリプトが終了することなく常駐し、ログをリアルタイムで確認することができないため、ログローテート機能を無効にすることを推奨します。監視タイプを [同期] に設定している場合、開始スクリプトから起動した常駐プロセスの標準出力を /dev/null へ指定してください。

カスタムモニタリソースの監視方法

カスタムモニタリソースは、任意のスクリプトによりシステム監視を行います。

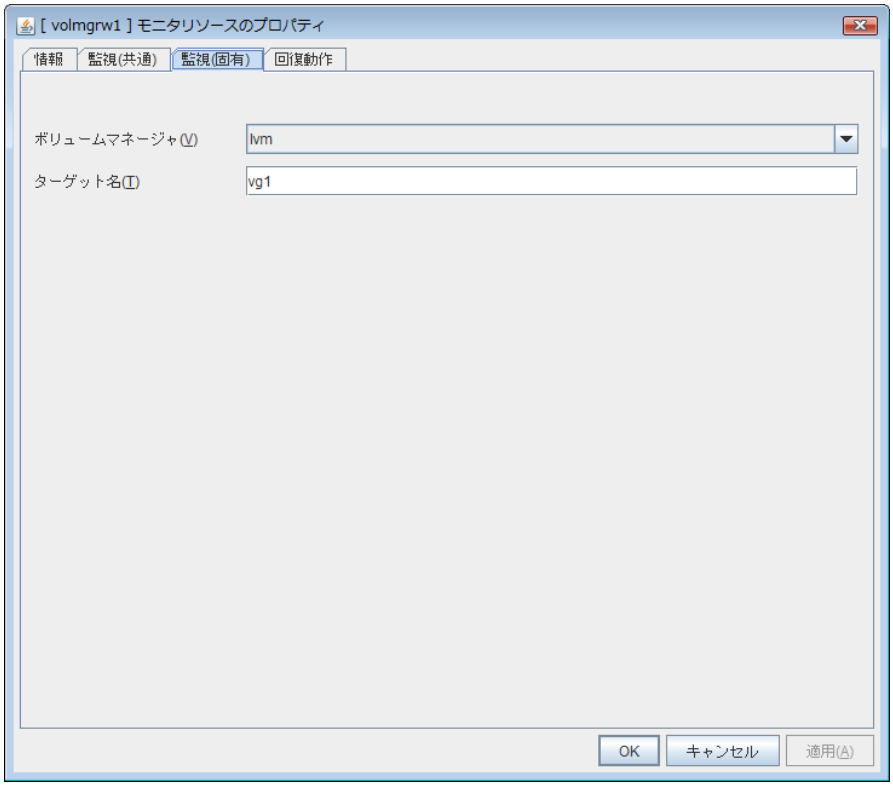
監視タイプが[同期]の場合、スクリプトを定期的に実行し、そのエラーコードにより異常の有無を判別します。

監視タイプが[非同期]の場合、スクリプトを監視開始時に実行し、このスクリプトのプロセスが消失した場合に異常と判断します。

ボリュームマネージャモニタリソースの設定

ボリュームマネージャモニタリソースは、ボリュームマネージャにより管理されている論理ディスクの監視を行うモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のボリュームマネージャモニタリソース名を右クリックし、[プロパティ] の [監視 (固有)] タブをクリックします。
3. [監視 (固有)] タブで、以下の説明に従い詳細設定の表示/変更を行います。



ボリュームマネージャ

監視対象の論理ディスクを管理しているボリュームマネージャの種類を設定します。対応済みのボリュームマネージャは下記です。

- lvm (LVM ボリュームグループ)
- vxvm (VxVM デーモン)

ターゲット名 (1023 バイト以内)

監視対象の名前を設定します。

ボリュームマネージャが [lvm] の場合、複数ボリュームをまとめて制御することができます。複数ボリュームを制御する場合は、ボリュームの名前を半角スペースで区切って設定します。

ボリュームマネージャが [vxvm] の場合、本設定の入力は不要です。

ボリュームマネージャモニタリソースの注意事項

ボリュームマネージャが VxVM の volmgrw は、デーモン監視形式のため、1 つのクラスタに複数登録しても意味はありません。

ボリュームマネージャに VxVM を指定した場合は、回復対象に LocalServer を設定してください。

ボリュームマネージャモニタリソースには既定値が設定されているため、必要があれば適切な値に変更してください。

ボリュームマネージャモニタリソースの監視方法

ボリュームマネージャモニタリソースは、監視する論理ディスクを管理するボリュームマネージャの種類によって監視方法が異なります。

対応済みのボリュームマネージャは下記です。

- lvm (LVM ボリュームグループ)
- vxvm (VxVM デーモン)

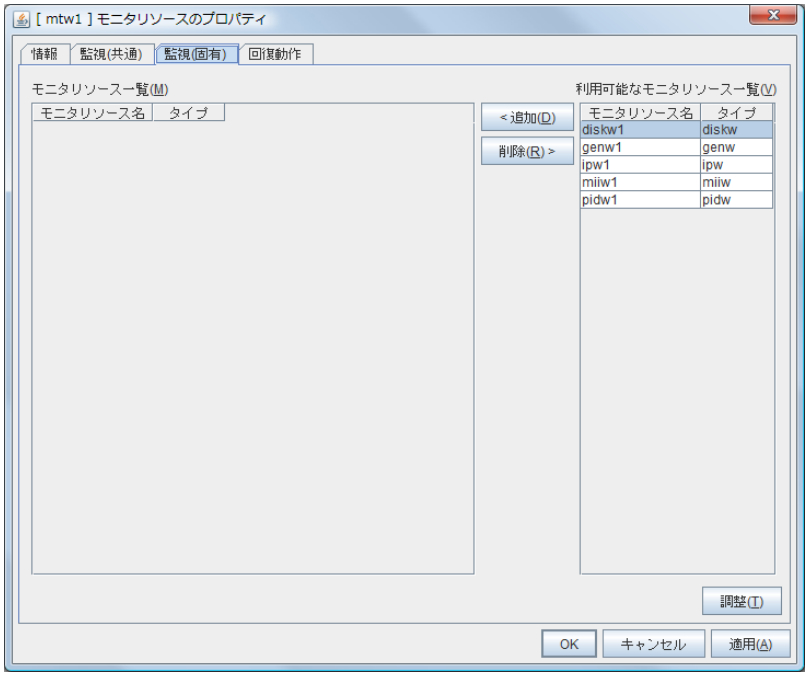
マルチターゲットモニタリソースの設定

マルチターゲットモニタリソースは、複数のモニタリソースの監視を行います。

モニタリソースをグループ化して、そのグループの状態を監視します。[モニタリソース一覧] はモニタリソースを最大 64 個登録できます。

本リソースの[モニタリソース一覧] に唯一設定されているモニタリソースが削除された場合、本リソースは自動的に削除されます。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のマルチターゲットモニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



追加

選択しているモニタリソースを[モニタリソース一覧] に追加します。

削除

選択しているモニタリソースを[モニタリソース一覧] から削除します。

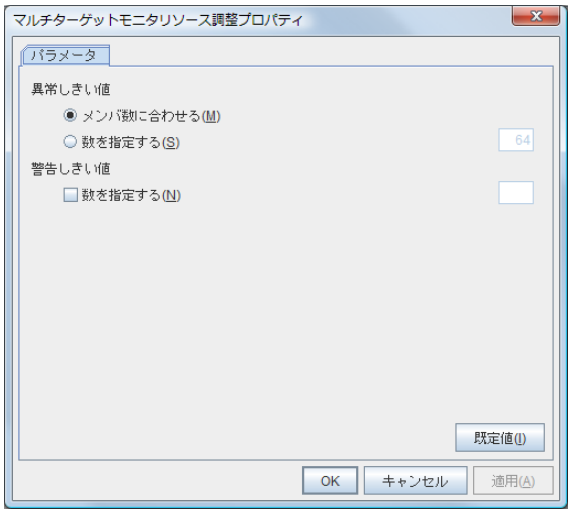
マルチターゲットモニタリソースの注意事項

- マルチターゲットモニタリソースは、登録されているモニタリソースのステータス 停止済み(offline)を異常として扱います。そのため、活性時監視のモニタリソースを登録した場合、モニタリソースが異常を検出していない状態でマルチターゲットモニタリソースが異常を検出してしまうことがあります。活性時監視のモニタリソースを登録しないでください。

マルチターゲットモニタリソースの調整を行うには

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のマルチターゲットモニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、[調整] をクリックします。[マルチターゲットモニタリソース調整プロパティ] ダイアログ ボックスが表示されます。
4. 以下の説明に従い詳細設定の表示 / 変更を行います。

パラメータタブ



異常しきい値

マルチターゲットモニタが異常とする条件を選択します。

- メンバ数に合わせる

  マルチターゲットモニタの配下に指定したモニタリソースが全て異常となったとき、または異常と停止済が混在しているときにマルチターゲットモニタが異常になります。

  マルチターゲットモニタの配下に指定したモニタリソースの全てが停止済の場合には、正常になります。

- 数を指定する

  マルチターゲットモニタの配下に指定したモニタリソースのうち、異常しきい値に設定した数が異常または停止済となったときにマルチターゲットモニタが異常になります。

マルチターゲットモニタの配下に指定したモニタリソースのうち、何個のモニタリソースが異常または停止済となったときにマルチターゲットモニタを異常とするかの個数を設定します。

異常しきい値の選択が[数を指定する]のときに設定できます。

警告しきい値

- チェックボックスがオン

  マルチターゲットモニタの配下に指定したモニタリソースのうち、何個のモニタリソースが異常または停止済となったときにマルチターゲットモニタを警告とするかの個数を設定します。

- チェックボックスがオフ

  マルチターゲットモニタは警告のアラートを表示しません。

既定値

既定値に戻すときに使用します。[既定値]ボタンを選択すると全ての項目に既定値が設定されます。

マルチターゲットモニタリソースのステータス

マルチターゲットモニタリソースのステータスは登録されているモニタリソースのステータスによって判断します。

マルチターゲットモニタリソースが下記のように設定されている場合、

登録されているモニタリソース数	2
異常しきい値	2
警告しきい値	1

マルチターゲットモニタリソースのステータスは以下のようになります。

マルチターゲット モニタリソースステータス		**モニタリソース1 ステータス**		
		正常 (normal)	異常 (error)	停止済 (offline)
モニタリソース2 ステータス	正常 (normal)	正常 (normal)	警告 (caution)	警告 (caution)
	異常 (error)	警告 (caution)	異常 (error)	異常 (error)
	停止済 (offline)	警告 (caution)	異常 (error)	正常 (normal)

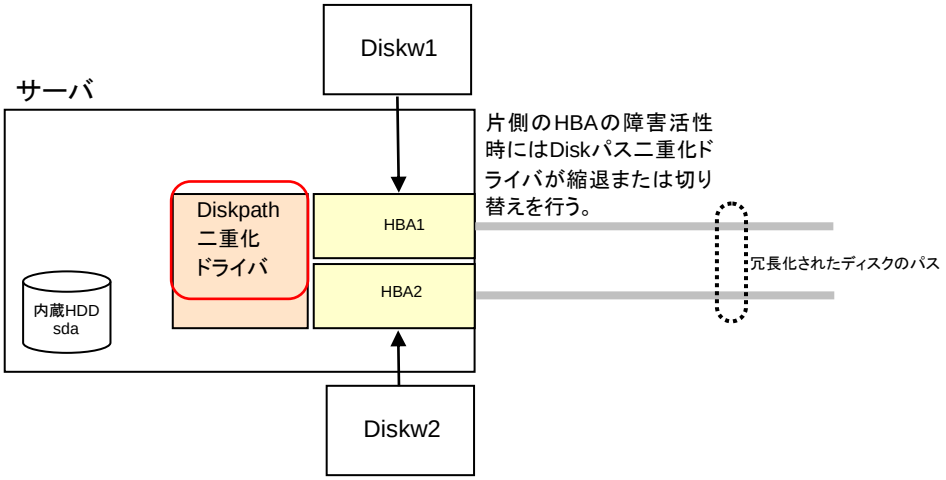
- マルチターゲットモニタリソースは、登録されているモニタリソースのステータスを監視しています。
  ステータスが異常(error)であるモニタリソースの数が異常しきい値以上になった場合、マルチターゲットモニタリソースは異常(error)を検出します。
  ステータスが異常(error)であるモニタリソース数が警告しきい値を超えた場合、マルチターゲットモニタリソースの status は警告(caution)となります。
  登録されている全てのモニタリソースのステータスが停止済み(offline)の場合、マルチターゲットモニタリソースのステータスは正常(normal)となります。

登録されている全てのモニタリソースのステータスが停止済み(offline)の場合を除いて、マルチターゲットモニタリソースは登録されているモニタリソースのステータス 停止済み(offline)を異常(error)と判断します。

- 登録されているモニタリソースのステータスが異常(error)となっても、そのモニタリソースの異常時アクションは実行されません。
  マルチターゲットモニタリソースが異常(error)になった場合のみ、マルチターゲットモニタリソースの異常時アクションが実行されます。

マルチターゲットモニタリソースの設定例

◆ Disk パス二重化ドライバの使用例
ディスクデバイス(/dev/sdb, /dev/sdc など)が同時に異常となった場合にのみ、異常(error)とする必要があります。



- マルチターゲットモニタリソース(mtw1)に登録するモニタリソース
  - diskw1
  - diskw2
- マルチターゲットモニタリソース(mtw1)の異常しきい値、警告しきい値
  - 異常しきい値 2
  - 警告しきい値 0
- マルチターゲットモニタリソース(mtw1)に登録するモニタリソースの詳細設定
  - ディスクモニタリソース(diskw1)
    監視デバイス名 /dev/sdb
    再活性しきい値 0
    フェイルオーバしきい値 0
    最終動作 何もしない
  - ディスクモニタリソース(diskw2)
    監視デバイス名 /dev/sdc
    再活性しきい値 0
    フェイルオーバしきい値 0
    最終動作 何もしない

◆ 上記の設定の場合、マルチターゲットモニタリソースのモニタリソースに登録されている diskw1 と diskw2 のどちらかが異常を検出しても、異常となったモニタリソースの異常時アクションを行いません。

◆ diskw1 と diskw2 が共に異常となった場合、2 つのモニタリソースのステータスが異常(error)と停止済み(offline)になった場合、マルチターゲットモニタリソースに設定された異常時アクションを実行します。

ソフト RAID モニタリソースの設定

ソフト RAID モニタリソースは、ソフト RAID を行っているデバイスを監視するモニタリソースです。

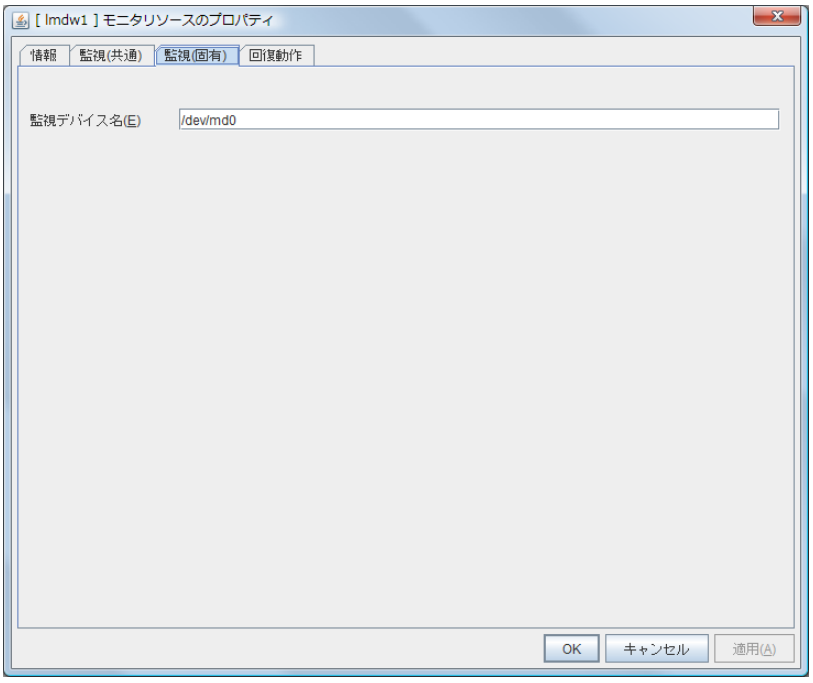
ソフトRAIDモニタリソースの監視方法

ソフト RAID モニタリソースは、md ドライバを利用してソフト RAID を行っているデバイスを監視します。片方の DISK が異常になって、ソフト RAID が縮退している場合に WARNING を通知します。

注意) 両方のディスクが異常になった場合は、異常を検出できませんので、縮退の通知時に DISK の復旧操作を行ってください。

ソフトRAIDモニタリソースの詳細を表示 / 変更するには

1. Builder の左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の ソフト RAID モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



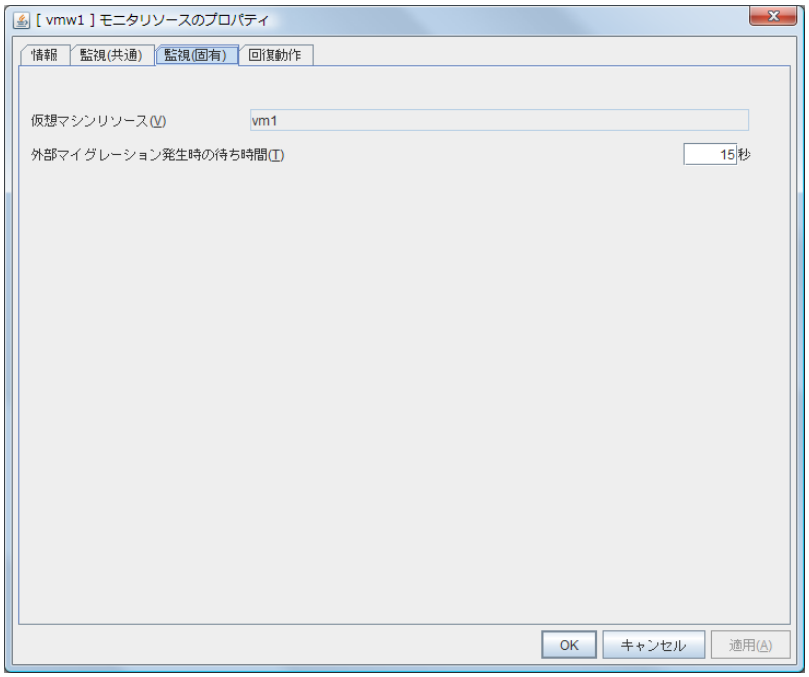
監視デバイス名(1023 バイト以内)

監視を行う md デバイス名を設定してください。

仮想マシンモニタリソースの設定

仮想マシンモニタリソースは、仮想マシンの生存確認を行うモニタリソースです。

1. Builder 左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の仮想マシンモニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います



外部マイグレーション発生時の待ち時間

マイグレーションが完了するまでにかかる時間を指定します。

仮想マシンモニタリソースの注意事項

- 動作確認済みの仮想マシンのバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

- 回復動作遷移中または全ての回復動作完了後に仮想マシンモニタリソースの復帰を検出しても、モニタリソースが保持している回復動作の回数カウンタはリセットされません。回復動作の回数カウンタをリセットしたい場合は、下記のいずれかの処理を実行してください。
  - clpmonctrl コマンドを使って、回復動作の回数カウンタをリセットする。
  - clpcl コマンドまたは Web マネージャから、クラスタ停止/開始を実行する。

仮想マシンモニタリソースの監視方法

仮想マシンモニタリソースは、以下の監視を行います。

仮想マシンの種類が vSphere の場合

VMware vSphere API を利用して仮想マシンの監視を行います。

監視の結果、以下の場合に異常とみなします。

(1) 仮想マシン の状態が POWEROFF/SHUTDOWN/SUSPENDED の場合

(2) 仮想マシン の状態取得に失敗した場合

仮想マシンの種類が Xenserver の場合

汎用の仮想化ライブラリを利用して仮想マシンの監視を行います。

監視の結果、以下の場合に異常とみなします。

(1) 仮想マシン の状態が HALTED/PAUSED/SUSPENDED の場合

(2) 仮想マシン の状態取得に失敗した場合

仮想マシンの種類が Kvm の場合

汎用の仮想化ライブラリを利用して仮想マシンの監視を行います。

監視の結果、以下の場合に異常とみなします。

(1) 仮想マシン の状態が

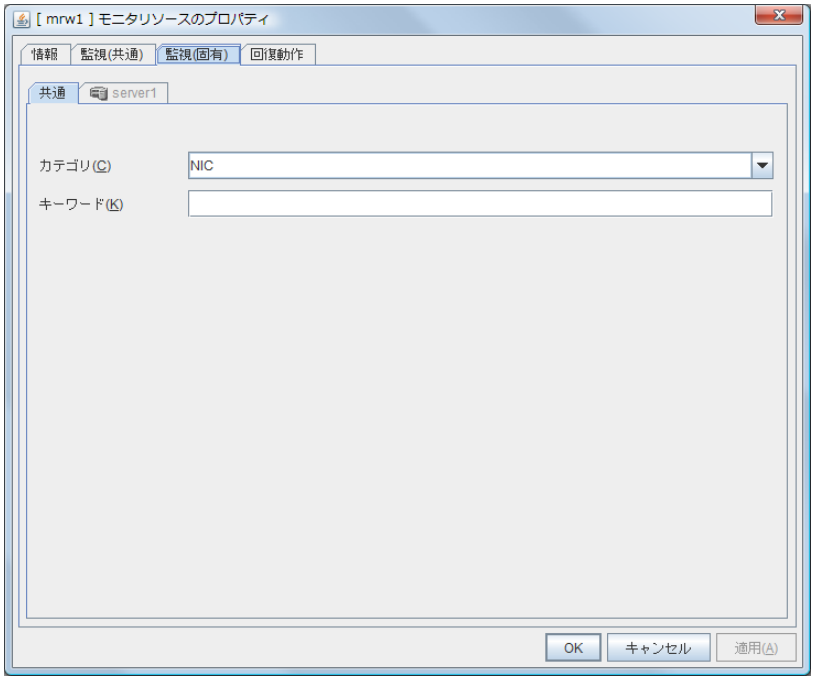
BLOCKED/SHUTDOWN/PAUSED/SHUTOFF/CRASHED/NOSTATE の場合

(2) 仮想マシン の状態取得に失敗した場合

外部連携モニタリソースの設定

外部連携モニタリソースは受動的なモニタです。自身では監視処理を行いません。
CLUSTERPRO の外部から発行された異常発生通知を受信した場合に、外部連携モニタリソースのステータスの変更、異常発生時の回復動作を行うモニタリソースです。

1. Builder 左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の外部連携モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



カテゴリとキーワードには、clprexec コマンドの引数-k で渡すキーワードを設定します。キーワードは省略可能です。

カテゴリ(32 バイト以内)

clprexec コマンドの引数-k で指定するカテゴリを指定します。
リストボックスでの既定文字列の選択または任意の文字列の指定が可能です。

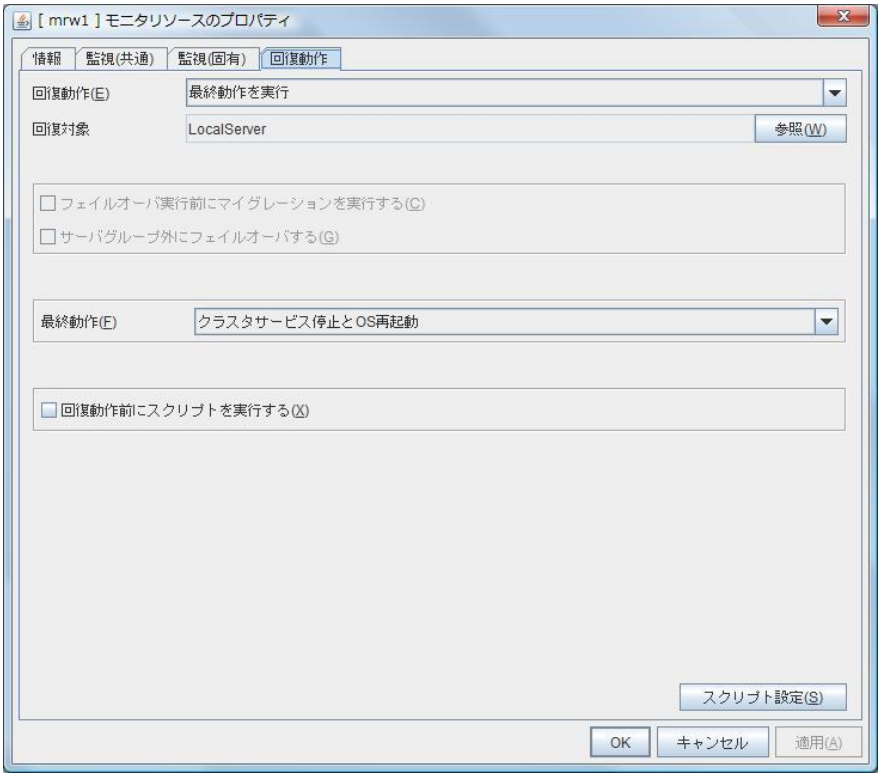
キーワード (1023 バイト以内)

clprexec コマンドの引数-k で指定するキーワードを指定します。

外部連携モニタリソースの異常検出時の設定

回復対象と異常検出時の動作を設定します。外部連携モニタリソースの場合、異常検出時の動作は、"回復対象を再起動"、"回復対象に対してフェイルオーバ実行"、または、"最終動作を実行" のいずれか1つを選択します。ただし、回復対象が非活性状態であれば回復動作は行われません。

1. Builder の左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のモニタリソース名を右クリックし、[プロパティ] の [回復動作] タブをクリックします。
3. [回復動作] タブで、以下の説明に従い監視設定の表示 / 変更を行います。



回復動作

モニタ異常検出時に行う動作を選択します。

- 回復スクリプトを実行
  モニタ異常検出時に、回復スクリプトを実行します。
- 回復対象を再起動
  モニタ異常検出時に、回復対象に選択したグループまたはグループリソースの再起動を行います。
- 最終動作を実行
  モニタ異常検出時に、最終動作に選択した動作を行います。

回復動作前にスクリプトを実行する

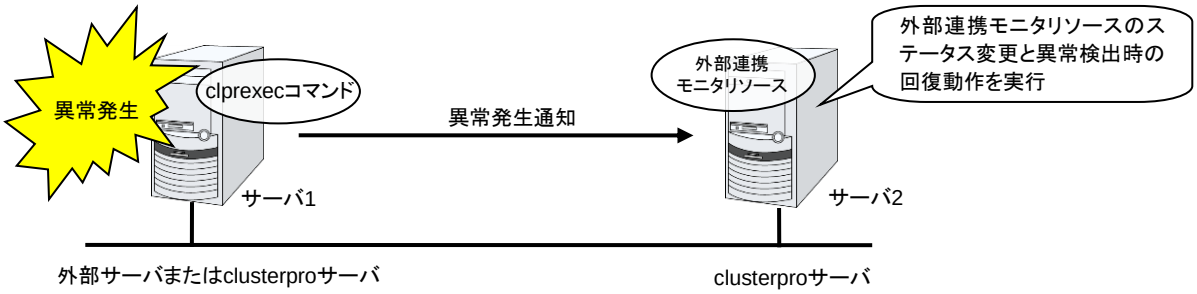
回復動作を実行する前にスクリプトを実行するかどうかを指定します。

- チェックボックスがオン
  回復動作を実施する前にスクリプト/コマンドを実行します。スクリプト/コマンドの設定を行うためには [スクリプト設定] をクリックしてください。
- チェックボックスがオフ
  スクリプト/コマンドを実行しません。

※ 上記以外の設定項目については、「第 5 章　モニタリソースの詳細」-「モニタリソース共通の設定」-「2. 復旧処理の設定」 を参照してください。

外部連携モニタリソースの監視方法

- 外部から異常発生通知を受信した場合、通知されたカテゴリとキーワード(キーワードは省略可能)が設定されている外部連携モニタリソースの異常発生時の回復動作を行います。通知されたカテゴリ、キーワードが設定されている外部連携モニタリソースが複数存在する場合は、各モニタリソースの回復動作を行います。



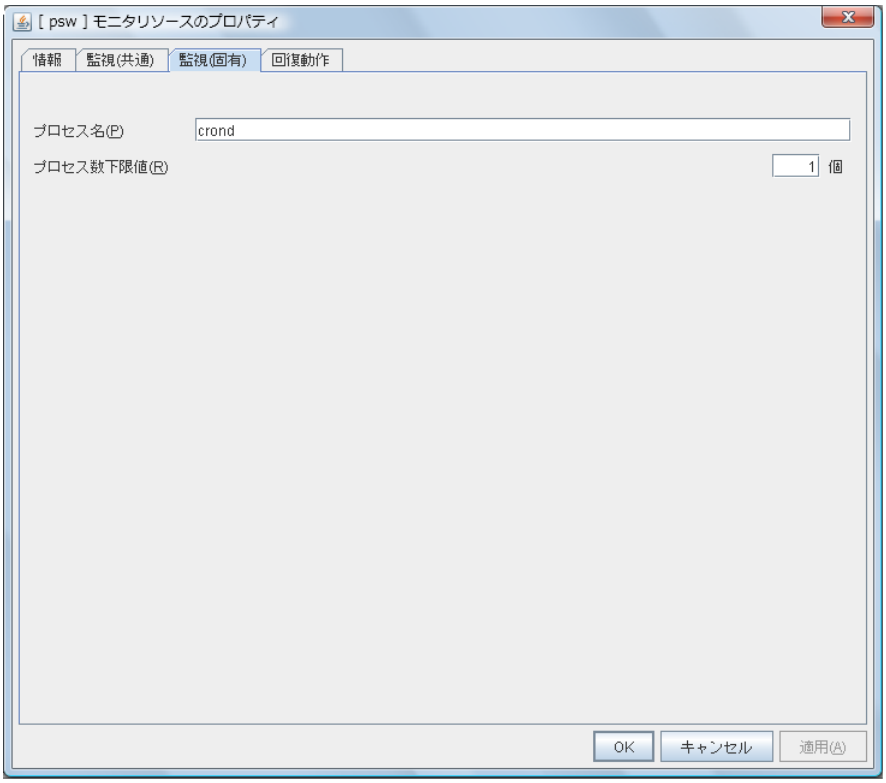
外部連携モニタリソースに関する注意事項

- 外部連携モニタリソースが一時停止状態で外部からの異常発生通知を受信した場合、異常時動作は実行されません。
- 外部から異常発生通知を受信した場合、外部連携モニタリソースのステータスは”異常”になります。“異常”となった外部連携モニタリソースのステータスは、自動では“正常”に戻りません。ステータスを”正常”に戻したい場合は、clprexecコマンドを使用してください。clprexecコマンドについては『操作ガイド』の「第 2 章 CLUSTERPRO X SingleServerSafeコマンドリファレンス」を参照してください。
- 外部から異常発生通知を受信して外部連携モニタリソースのステータスが"異常"となっている状態で異常発生通知を受信した場合、異常発生時の回復動作は実行されません。

プロセス名モニタリソースの設定

プロセス名モニタリソースは、任意のプロセス名のプロセスを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のプロセス名モニタリソース名を右クリックし、[プロパティ] の [監視 (固有)] タブをクリックします。
3. [監視 (固有)] タブで、以下の説明に従い詳細設定の表示/変更を行います。



プロセス名 (1023 バイト以内)

監視対象プロセスのプロセス名を設定します。プロセス名は ps(1)コマンドの出力結果などから確認します。

また、次の3つのパターンでプロセス名のワイルドカード指定が可能です。このパターン以外の指定はできません。

【前方一致】　＜プロセス名に含まれる文字列＞＊

【後方一致】　＊＜プロセス名に含まれる文字列＞

【部分一致】　＊＜プロセス名に含まれる文字列＞＊

プロセス数下限値（1～999）

監視対象プロセスの監視個数を設定します。プロセス名に設定した監視対象プロセスの個数が設定値を下回った場合に異常と判断します。

プロセス名モニタリソースの注意事項

プロセス数下限値に 1 を設定した場合に監視対象に指定したプロセス名のプロセスが複数存在すると、次の条件で監視対象プロセスを一つ選択し監視します。

1. プロセス間に親子関係がある場合は、親プロセスを監視します。
2. プロセス間に親子関係がなければ、プロセスの起動時刻の最も古いものを監視します。
3. プロセス間に親子関係がなく、プロセスの起動時刻も同じであれば、もっともプロセスIDの小さいものを監視します。

同一名のプロセスが複数存在する場合にプロセスの起動個数によって監視を行う際には、プロセス数下限値に監視する個数を設定します。同一名のプロセスが設定された個数を下回ると異常と判断します。プロセス数下限値に指定できる個数は 1 から 999 個までです。プロセス数下限値に1を設定した場合は、監視対象プロセスを一つ選択して監視します。

監視対象プロセス名に指定できるプロセス名は1023バイトまでです。1023バイトを超えるプロセス名を持つプロセスを監視対象として指定する場合は、ワイルドカード（＊）を使って指定します。

監視対象プロセスのプロセス名が1023バイトより長い場合、プロセス名として認識できるのはプロセス名の先頭から1023バイトまでです。ワイルドカード（＊）を使って指定する場合は、1024バイトまでに含まれる文字列を指定してください。

監視対象のプロセス名が長い場合、ログ等に出力されるプロセス名情報は後半を省略して表示されます。

プロセス名の中に「“」（ダブルコーテーション）や「,」（カンマ）が含まれるプロセスを監視している場合、アラートメッセージにプロセス名が正しく表示できない場合があります。

監視対象プロセス名は、実際に動作しているプロセスのプロセス名を ps(1)コマンド等で確認し設定してください。

実行結果の例

```
# ps -eaf
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 Sep12 ?        00:00:00 init [5]
  :
root        5314       1  0 Sep12 ?        00:00:00 /usr/sbin/acpid
root        5325       1  0 Sep12 ?        00:00:00 /usr/sbin/sshd
htt         5481       1  0 Sep12 ?        00:00:00 /usr/sbin/htt -retryonerror 0
  :
```

上記のコマンド実行結果から /usr/sbin/htt を監視する場合、
/usr/sbin/htt -retryonerror 0 を監視対象プロセス名に指定します。

監視対象プロセス名に指定したプロセス名はプロセスの引数もプロセス名の一部として監視対象のプロセスを特定します。監視対象プロセス名を指定する場合は、引数を含めたプロセス名を指定してください。引数を含めずプロセス名のみ監視したい場合は、ワイルドカード（＊）を使い、引数を含めない前方一致または部分一致で指定してください。

プロセス名モニタリソースの監視方法

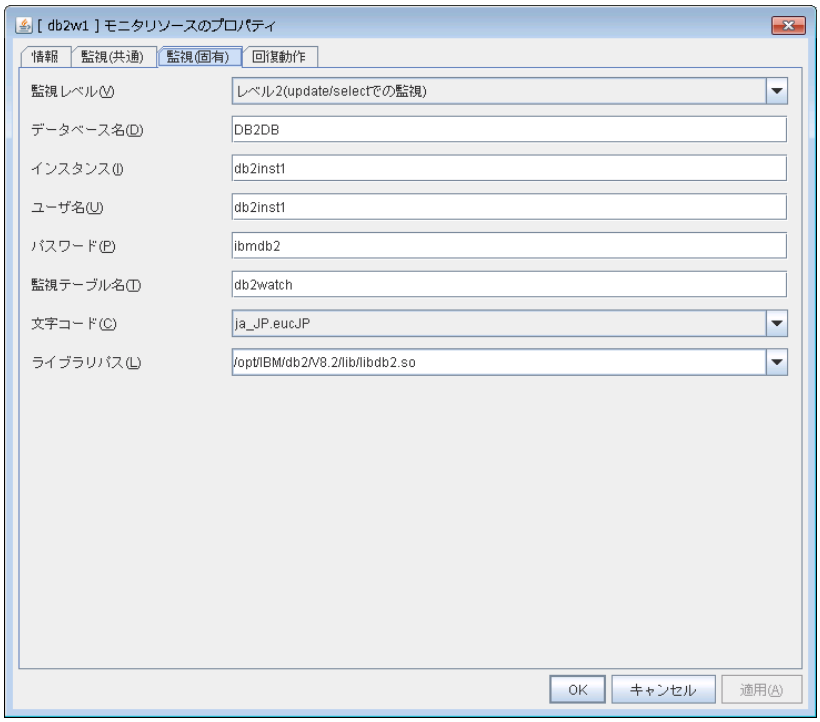
指定されたプロセス名のプロセスを監視します。プロセス数下限値に1を設定した場合、プロセス名からプロセス ID を特定し、そのプロセス ID の消滅時に異常と判断します。プロセスのストールを検出することはできません。

プロセス数下限値に1より大きい値を設定した場合、指定されたプロセス名のプロセスを個数によって監視します。プロセス名から監視対象プロセスの個数を算出し、下限値を下回った場合に異常と判断します。プロセスのストールを検出することはできません。

DB2 モニタリソースの設定

DB2 モニタリソースは、サーバ上で動作する DB2 のデータベースを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の DB2 モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



監視レベル

選択肢の中から1つを選択します。必ず設定してください。

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は(select)です。

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。監視テーブルに対して実行する SQL 文は(update / select)です。
  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は(create / insert)です。

- レベル 3(毎回 create/drop も行う)

  監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。監視テーブルに対して実行する SQL 文は(create / insert / select / drop)です。

既定値 : レベル2(update/selectでの監視)

データベース名(255 バイト以内)

監視するデータベース名を設定します。必ず設定してください。

既定値 : なし

インスタンス名(255 バイト以内)

監視するデータベースのインスタンス名を設定します。必ず設定してください。

既定値 : db2inst1

ユーザ名(255 バイト以内)

データベースにログインする際のユーザ名を設定します。必ず設定してください。

指定したデータベースにアクセス可能な DB2 ユーザを指定してください。

既定値 : db2inst1

パスワード(255 バイト以内)

データベースにログインする際のパスワードを設定します。必ず設定してください。

既定値 : ibmdb2

監視テーブル名(255 バイト以内)

データベース上に作成する監視用テーブルの名前を設定します。必ず設定してください。

テーブルの作成・削除を行いますので、運用に使用しているテーブル名と重ならないように注意してください。また、SQL 文の予約語と重ならないようにしてください。
データベースの仕様により監視テーブル名に設定できない文字があります。詳細はデータベースの仕様を確認してください。

既定値 : db2watch

文字コード

DB2 のキャラクタ・セットを設定します。必ず設定してください。

既定値 : なし

ライブラリパス(1023 バイト以内)

DB2 のライブラリパスを設定します。必ず設定してください。

既定値 : /opt/IBM/db2/V8.2/lib/libdb2.so

DB2モニタリソースの注意事項

動作確認済みの DB2 のバージョンについては『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースは、DB2 の CLI のライブラリを利用して、DB2 の監視を行っています。本モニタリソースが異常になる場合は、指定した DB2 の CLI のライブラリパスが存在することを確認してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する DB2 データベースを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 DB2 データベースが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。また、この場合は監視リソースが動作するホスト OS 側に DB2 クライアントをセットアップし、仮想マシン上のデータベースをデータベース ノード ディレクトリに登録しておく必要があります。

データベースのコードページと本モニタリソースの「文字コード」の設定が異なると、本モニタリソースは、DB2 のデータベースに接続することができません。必要に応じて、適切な文字コードの設定を行ってください。
データベースのコードページの確認は、「db2 get db cfg for データベース名」などで行ってください。詳細は、DB2 のマニュアルを参照してください。

パラメータで指定したデータベース名・インスタンス名・ユーザ名・パスワードなどの値が、監視を行う DB2 の環境と異なる場合、DB2 の監視を行うことができません。各エラー内容を示すメッセージが表示されますので、環境を確認してください。

次項の「DB2 モニタリソースの監視方法」で説明する監視レベルについて、「レベル1」「レベル2」を選択した場合、手動にて監視テーブルを作成しておく必要があります。
「レベル 1」で監視開始時に監視テーブルが無い場合、監視エラーになります。
「レベル 2」で監視開始時に監視テーブルが無い場合、CLUSTERPRO が監視テーブルを自動で作成します。このとき WebManager のアラートビューに監視テーブルがない旨のメッセージが表示されます。
「レベル 3」での監視は毎回監視テーブルの作成・削除を行うため「レベル 1」「レベル 2」より監視の負荷が高くなります。

選択する監視レベル	監視テーブルの事前作成
レベル1（selectでの監視）	必要あり
レベル2（update/selectでの監視）	必要あり
レベル3（毎回create/dropも行う）	必要なし

監視テーブルの作成は以下の手順で行えます。

SQL 文で作成する場合（以下の例は監視テーブル名を db2watch とする場合）
sql> create table <ユーザ名>.db2watch (num int not null primary key)
sql> insert into db2watch values(0)
sql> commit

CLUSTERPRO のコマンドを利用する場合
clp_db2w --createtable -n <DB2 モニタリソース名>
作成した監視テーブルを手動で削除する場合、下記のコマンドを実行してください:
clp_db2w --deletetable -n <DB2 モニタリソース名>

DB2モニタリソースの監視方法

DB2 モニタリソースは、以下の監視レベルから選択した監視レベルに応じた監視を行います。

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（ select ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ update / select ）です。

  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

- レベル 3(毎回 create/drop も行う)

  監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ create / insert / select / drop ）です。

  監視の結果以下の場合に異常とみなします。

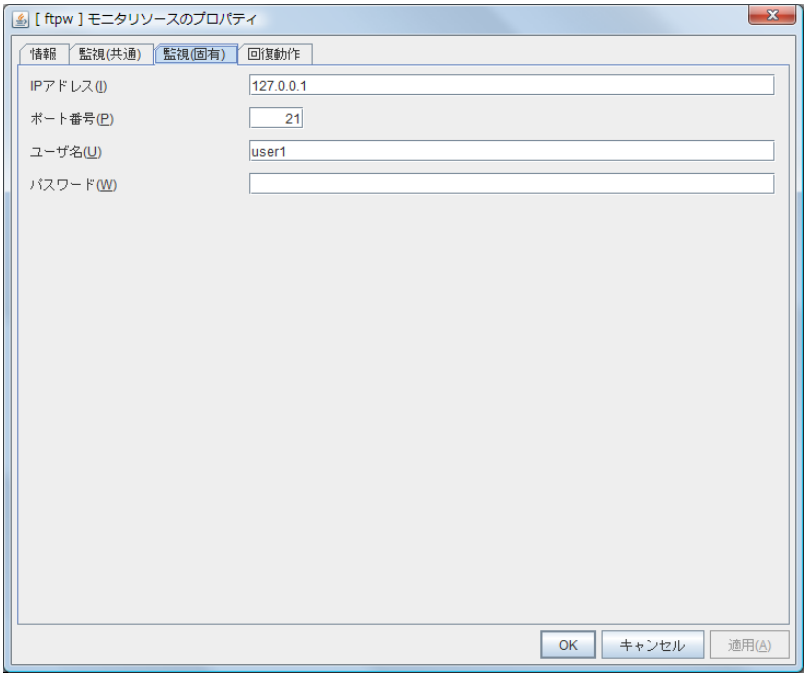
  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

FTP モニタリソースの設定

FTP モニタリソースは、サーバ上で動作する FTP サービス監視するモニタリソースです。FTP プロトコルを監視するものであり、特定のアプリケーションの監視ではありません。そのため、FTP プロトコルを実装するさまざまなアプリケーションの監視を行うことができます。

1. Builder 左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の FTP 監視リソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



IP アドレス (79 バイト以内)

監視する FTP サーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する FTP サーバに接続しますので、ループバックアドレス(127.0.0.1)を設定しますが、FTP サーバの設定で接続可能なアドレスを制限している場合は、接続可能なアドレス(フローティング IP アドレス等)を設定します。また、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する FTP サーバを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 : 127.0.0.1

ポート番号 (1～65535)

監視する FTP のポート番号を設定します。必ず設定してください。

既定値 ： 21

ユーザ名(255 バイト以内)

FTP にログインする際のユーザ名を設定します。

既定値 ： なし

パスワード(255 バイト以内)

FTP にログインする際のパスワードを設定します。

既定値 ： なし

FTPモニタリソースの注意事項

監視の対象リソースには、FTPを起動するexecリソースなどを指定してください。対象リソースの活性後、監視を開始しますが、対象リソースの活性直後に FTP がすぐに動作できない場合などは、[監視開始待ち時間]で調整してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する FTP サーバを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 FTP サーバが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。

監視動作ごとにFTP サービス自体が動作ログなどを出力することがありますが、その制御は、FTP 側の設定で適宜行ってください。

FTP サーバの FTP メッセージ(バナー、ウェルカムメッセージなど)を既定から変更すると、監視異常とみなす場合があります。

FTPモニタリソースの監視方法

FTP モニタリソースは、以下の監視を行います。

FTP サーバに接続してファイル一覧取得コマンドを実行します。

監視の結果、以下の場合に異常とみなします。

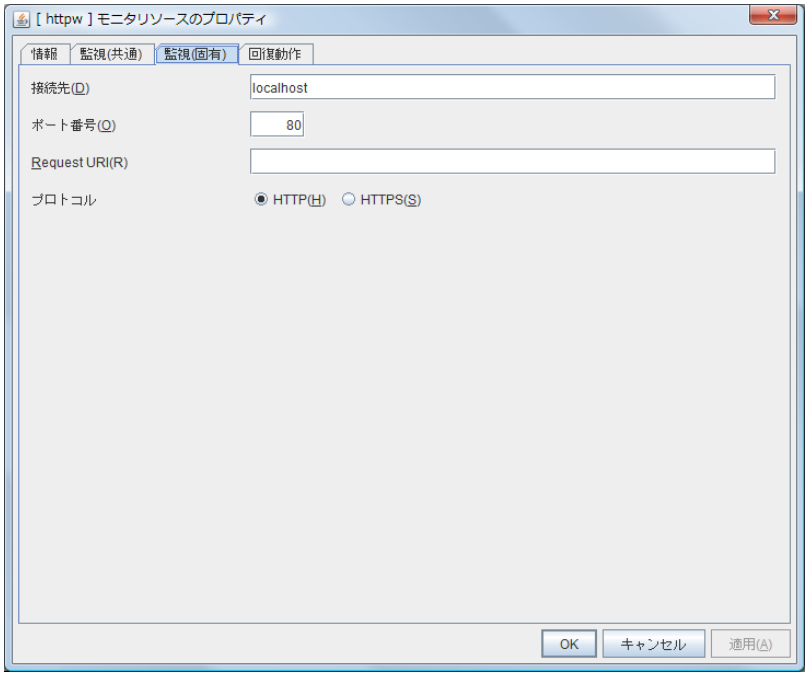
(1) FTP サービスへの接続に失敗した場合

(2) FTP コマンドに対する応答で異常が通知された場合

HTTP モニタリソースの設定

HTTP モニタリソースは、サーバ上で動作する HTTP デーモンを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の HTTP モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



接続先(255 バイト以内)

監視する HTTP サーバ名を設定します。必ず設定してください。

通常は自サーバ上で動作する HTTP サーバに接続しますので、ループバックアドレス(127.0.0.1)を設定しますが、HTTP サーバの設定で接続可能なアドレスを制限している場合は、接続可能なアドレスを設定します。また、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する HTTP サーバを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 : localhost

ポート番号(1～65535)

HTTP サーバに接続する際のポート番号を設定します。必ず設定してください。

既定値 : 80 (HTTP の場合)
443 (HTTPS の場合)

Request URI(255 バイト以内)

Request URI（例：”/index.html”）を設定します。

既定値 ： なし

プロトコル

HTTP サーバとの通信に使用するプロトコルを設定します。通常は HTTP を選択しますが、HTTP over SSL で接続する必要がある場合は HTTPS を選択します。

既定値 ： HTTP

HTTPモニタリソースの注意事項

動作確認済みの HTTP のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する HTTP サーバを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 HTTP サーバが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。

HTTP モニタリソースはクライアント認証、BASIC 認証、DIGEST 認証に未対応です。

HTTPモニタリソースの監視方法

HTTP モニタリソースは、以下の監視を行います。

サーバ上の HTTP デーモンに接続し、HEAD リクエストの発行により、HTTP デーモンの監視を実行します。

監視の結果、以下の場合に異常とみなします。

(1) HTTP デーモンへの接続で異常が通知された場合

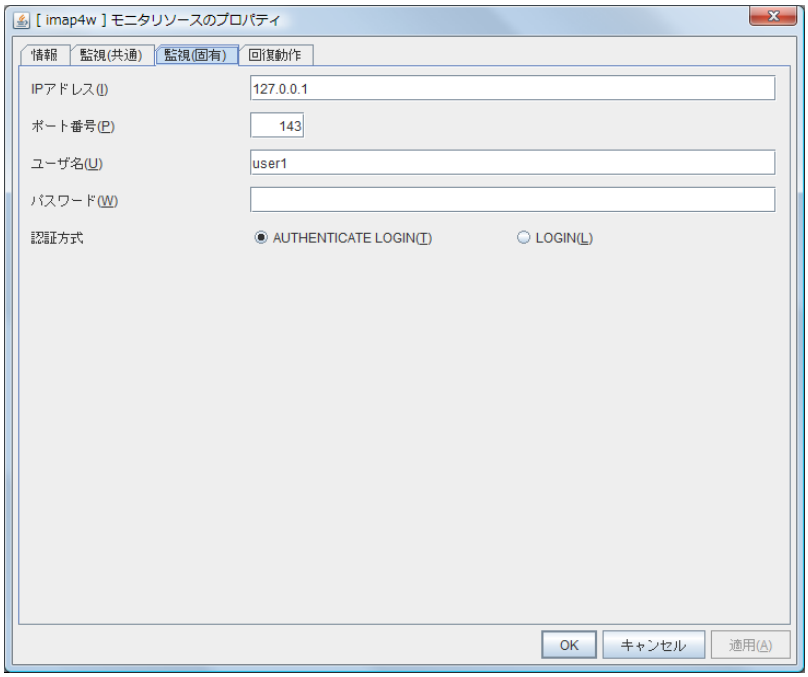
(2) HEAD リクエストの発行に対する応答メッセージが "HTTP/" で始まっていない場合

(3) HEAD リクエストの発行に対する応答のステータスコードが 400、500 番台の場合（Request URI に既定値以外の URI を指定した場合）

IMAP4 モニタリソースの設定

IMAP4 モニタリソースは、サーバ上で動作する IMAP4 のサービスを監視するモニタリソースです。IMAP4 プロトコルを監視するものであり、特定のアプリケーションの監視ではありません。そのため、IMAP4 プロトコルを実装するさまざまなアプリケーションの監視を行うことができます。

1. Builder 左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の IMAP4 モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



IP アドレス (79 バイト以内)

監視する IMAP4 サーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する IMAP4 サーバに接続しますので、ループバックアドレス(127.0.0.1)を設定しますが、IMAP4 サーバの設定で接続可能なアドレスを制限している場合は、接続可能なアドレスを設定します。また、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する IMAP4 サーバを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 : 127.0.0.1

ポート番号 (1～65535)

監視する IMAP4 のポート番号を設定します。必ず設定してください。

既定値 ： 143

ユーザ名(255 バイト以内)

IMAP4 にログインする際のユーザ名を設定します。

既定値 ： なし

パスワード(255 バイト以内)

IMAP4 にログインする際のパスワードを設定します。[変更]ボタンを押してパスワード指定ダイアログを表示して設定します。

既定値 ： なし

認証方式

IMAP4 にログインするときの認証方式を選択します。使用している IMAP4 の設定に合わせる必要があります。

- AUTHENTICATE LOGIN(既定値)

  AUTHENTICATE LOGIN コマンドを使用した暗号化認証方式です。

- LOGIN

  LOGIN コマンドを使用した平文方式です。

IMAP4モニタリソースの注意事項

監視の対象リソースには、IMAP4 サーバを起動する exec リソースなどを指定してください。対象リソースの活性後、監視を開始しますが、対象リソースの活性直後に IMAP4 サーバがすぐに動作できない場合などは、[監視開始待ち時間]で調整してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する IMAP4 サーバを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 IMAP4 サーバが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。

監視動作ごとに IMAP4 サーバ自体が動作ログなどを出力することがありますが、その制御は、IMAP4 サーバ側の設定で適宜行ってください。

IMAP4モニタリソースの監視方法

IMAP4 モニタリソースは、以下の監視を行います。

IMAP4 サーバに接続して動作確認コマンドを実行します。

監視の結果、以下の場合に異常とみなします。

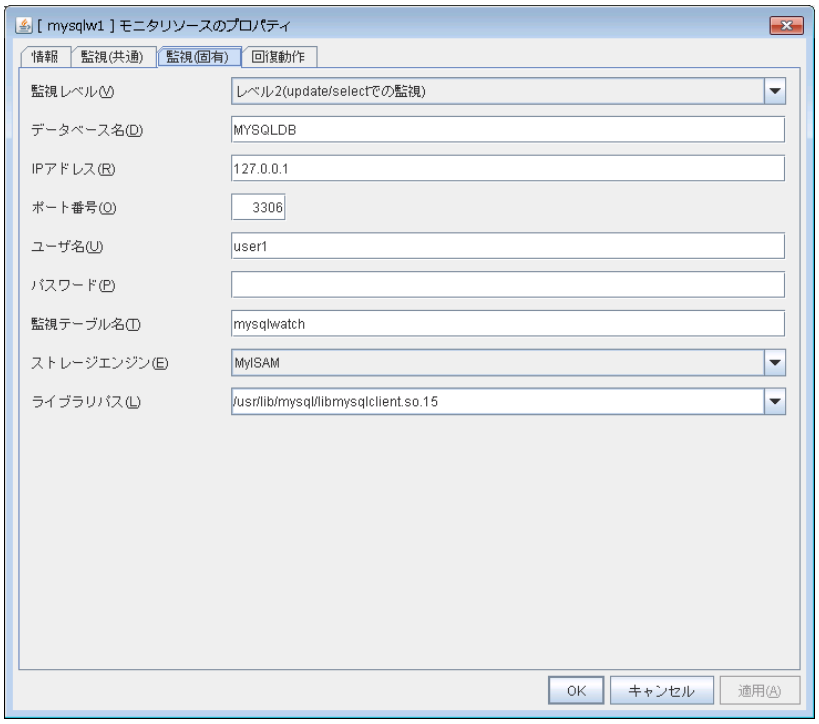
(1) IMAP4 サーバへの接続に失敗した場合

(2) コマンドに対する応答で異常が通知された場合

MySQL モニタリソースの設定

MySQL モニタリソースは、サーバ上で動作する MySQL のデータベースを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の MySQL モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



監視レベル

選択肢の中から1つを選択します。必ず設定してください。

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（ select ）です。

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。監視テーブルに対して実行する SQL 文は（ update / select ）です。
  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

- レベル 3(毎回 create/drop も行う)

監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。監視テーブルに対して実行する SQL 文は（ create / insert / select / drop ）です。

既定値 ： レベル2(update/selectでの監視)

データベース名(255 バイト以内)

監視するデータベース名を設定します。必ず設定してください。

既定値 ： なし

IP アドレス(79 バイト以内)

接続するサーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する MySQL サーバに接続しますので、ループバックアドレス（127.0.0.1）を設定しますが、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する MySQL データベースを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 ： 127.0.0.1

ポート番号(1～65535)

接続する際のポート番号を設定します。必ず設定してください。

既定値 ： 3306

ユーザ名(255 バイト以内)

データベースにログインする際のユーザ名を設定します。必ず設定してください。

指定したデータベースにアクセス可能な MySQL ユーザを指定してください。

既定値 ： なし

パスワード(255 バイト以内)

データベースにログインする際のパスワードを設定します。

既定値 ： なし

監視テーブル名(255 バイト以内)

データベース上に作成する監視用テーブルの名前を設定します。必ず設定してください。

テーブルの作成・削除を行いますので、運用に使用しているテーブル名と重ならないように注意してください。また、SQL 文の予約語と重ならないようにしてください。
データベースの仕様により監視テーブル名に設定できない文字があります。詳細はデータベースの仕様を確認してください。

既定値 ： mysqlwatch

ストレージエンジン

監視用 Table の作成用ストレージエンジンを設定します。必ず設定してください。

既定値 ： MyISAM

ライブラリパス(1023 バイト以内)

MySQL のライブラリパスを設定します。必ず設定してください。

既定値 ： /usr/lib/mysql/libmysqlclient.so.15

MySQLモニタリソースの注意事項

動作確認済みの MySQL のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースは、MySQL の libmysqlclient のライブラリを利用して、MySQL の監視を行っています。
本モニタリソースが異常になる場合は、MySQL のライブラリのインストールディレクトリに libmysqlclient.so.xx が存在することを確認してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する MySQL データベースを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 MySQL データベースが接続可能となるまでの十分な待ち時間を[監視開始待ち時間]に設定してください。

パラメータ指定値が、監視を行う MySQL の環境と異なる場合、WebManager のアラートビューに、エラー内容を示すメッセージが表示されますので、環境を確認してください。

次項の「MySQL モニタリソースの監視方法」で説明する監視レベルについて、「レベル 1」「レベル 2」を選択した場合、手動にて監視テーブルを作成しておく必要があります。
「レベル 1」で監視開始時に監視テーブルが無い場合、監視エラーになります。
「レベル 2」で監視開始時に監視テーブルが無い場合、CLUSTERPRO が監視テーブルを自動で作成します。このとき WebManager のアラートビューに監視テーブルがない旨のメッセージが表示されます。
「レベル 3」での監視は毎回監視テーブルの作成・削除を行うため「レベル 1」「レベル 2」より監視の負荷が高くなります。

選択する監視レベル	監視テーブルの事前作成
レベル1(selectでの監視)	必要あり
レベル2(update/selectでの監視)	必要あり
レベル3(毎回create/dropも行う)	必要なし

監視テーブルの作成は以下の手順で行えます。

SQL 文で作成する場合（以下の例は監視テーブル名を mysqlwatch とする場合）
sql> create table mysqlwatch (num int not null primary key) ENGINE=<エンジン>;
sql> insert into mysqlwatch values(0);
sql> commit;

CLUSTERPRO のコマンドを利用する場合
clp_mysqlw --createtable -n <MySQL モニタリソース名>
作成した監視テーブルを手動で削除する場合、下記のコマンドを実行してください：
clp_mysqlw --deletetable -n <MySQL モニタリソース名>

MySQLモニタリソースの監視方法

MySQL モニタリソースは、以下の監視レベルから選択した監視レベルに応じた監視を行います。

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（ select ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ update / select ）です。

  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

- レベル 3(毎回 create/drop も行う)

  監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ create / insert / select / drop ）です。

  監視の結果以下の場合に異常とみなします。

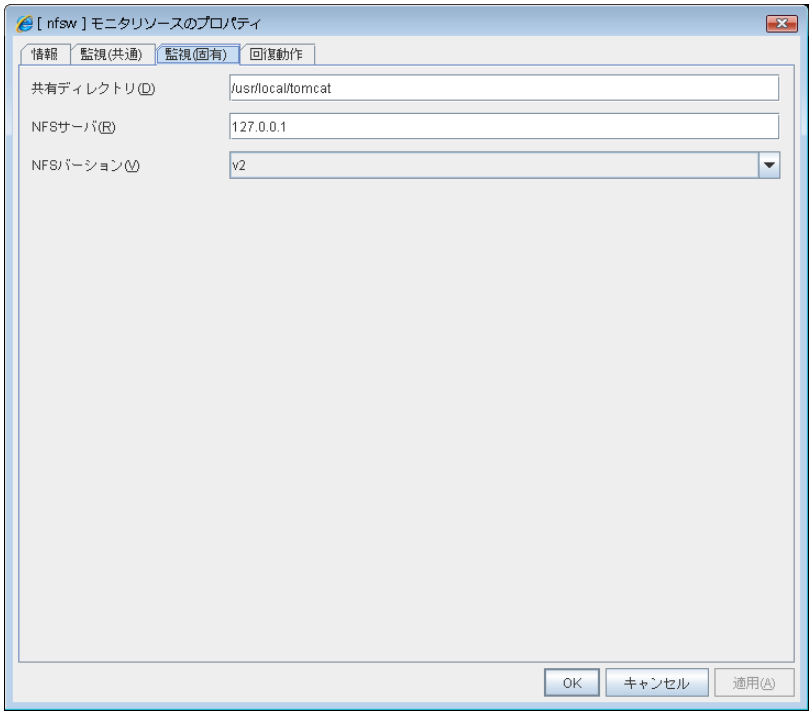
  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

NFS モニタリソースの設定

NFS モニタリソースは、サーバ上で動作する NFS のファイルサーバを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の NFS モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



共有ディレクトリ (1023 バイト以内)

ファイル共有するディレクトリを設定します。必ず設定してください。

既定値 : なし

NFS サーバ(79 バイト以内)

NFS 監視を行うサーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する NFS のファイルサーバに接続しますので、ループバックアドレス(127.0.0.1)を設定しますが、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する NFS のファイルサーバを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 : 127.0.0.1

NFS バージョン

NFS 監視を行う NFS のバージョンを選択肢の中から1つ選択します。必ず設定してください。
RHEL 7 では NFS バージョン v2 はサポートされていません。

- v2

  NFS バージョン v2 を監視します。

- v3

  NFS バージョン v3 を監視します。

- v4

  NFS バージョン v4 を監視します。

既定値 : v2

NFS モニタリソースの動作環境

NFS モニタリソースを利用するためには、以下のサービスが起動している必要があります。

<Red Hat Enterprise Linux 5 の場合>

- nfs
- portmap
- nfslock (NFS v4 では不要)

<Red Hat Enterprise Linux 6 の場合>

- nfs
- rpcbind
- nfslock (NFS v4 では不要)

NFSモニタリソースの注意事項

動作確認済みの NFS のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

監視する共有ディレクトリについては、自サーバから接続できるように exports ファイルを設定してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する NFS のファイルサーバを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 NFS のファイルサーバが接続可能となるまでの十分な待ち時間を[監視開始待ち時間]に設定してください。

[監視(固有)]タブ-[NFS バージョン]で指定したバージョンの nfsd およびその nfsd に対応する mountd の消滅を検出すると異常とみなします。nfsd に対応する mountd は以下の通りです。

nfsdバージョン	mountdバージョン
v2(udp)	v1(tcp)あるいはv2(tcp)
v3(udp)	v3(tcp)
v4(tcp)	—

NFSモニタリソースの監視方法

NFS モニタリソースは、以下の監視を行います。

NFS サーバに接続して NFS テストコマンドを実行します。

監視の結果、以下の場合に異常とみなします。

<Red Hat Enterprise Linux 5 の場合>

(1) NFS サービスへの要求に対する応答結果が異常な場合

(2) mountd が消滅した場合(NFS v4 を除く)

(3) nfsd が消滅した場合

(4) portmap サービスが停止した場合

(5) export されている領域が消滅した場合(NFS v4 を除く)

<Red Hat Enterprise Linux 6 の場合>

(1) NFS サービスへの要求に対する応答結果が異常な場合

(2) mountd が消滅した場合(NFS v4 を除く)

(3) nfsd が消滅した場合

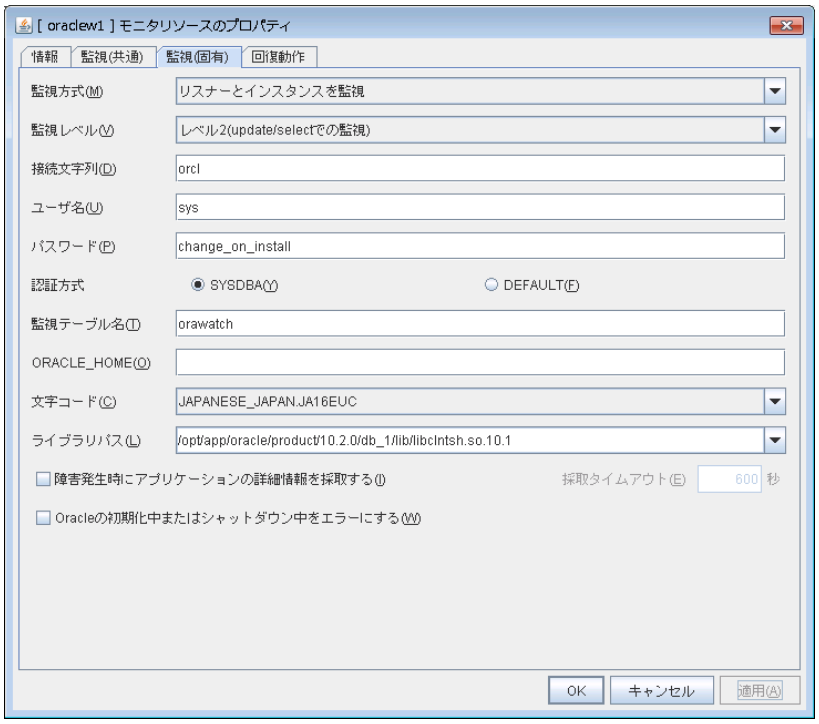
(4) rpcbind サービスが停止した場合

(5) export されている領域が消滅した場合(NFS v4 を除く)

Oracle モニタリソースの設定

Oracle モニタリソースは、サーバ上で動作する Oracle のデータベースを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の Oracle モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



監視方式

監視対象とする Oracle の機能を選択します。

- リスナーとインスタンスを監視(既定値)

  監視レベルに設定されたレベルに応じ、データベースへの接続、参照、更新の動作を監視します。

- リスナーのみ監視

  リスナーが動作しているかを Oracle のコマンド(tnsping)を実行し監視します。モニタリソースプロパティで ORACLE_HOME を設定しておく必要があります。

  ORACLE_HOME が設定されていない場合、接続文字列に指定されている先に対して接続処理の動作のみ監視します。接続異常時にリスナーのサービス再起動による復旧を試みる場合に使用します。

  本設定を選択した場合、監視レベルの設定は無視されます。

◆ インスタンスのみ監視

データベースに対しリスナーを経由せず直接接続（BEQ 接続）を行い、監視レベルに設定されたレベルに応じ、データベースへの接続、参照、更新の動作を監視します。モニタリソースプロパティで ORACLE_HOME を設定しておく必要があります。この方式はリスナーを経由せずインスタンスを直接監視し復旧動作を設定するために使用します。

監視対象が Oracle12c のマルチテナント構成のデータベースの場合、BEQ 接続での監視はできません。

ORACLE_HOME が設定されていない場合、接続文字列に指定されている先に対して接続を行い、接続処理で異常があった場合は無視します。この方式は、[リスナーのみ監視]方式の Oracle 監視リソースと併用して、接続処理以外の異常に対する復旧動作を設定するために使用します。

監視レベル

選択肢の中から1つを選択します。監視方式を「リスナーのみ監視」としている場合には本設定は無視されます。

◆ レベル 0(データベースステータス)

Oracle の管理テーブル（V$INSTANCE 表）を参照し DB の状態（インスタンスの状態）を確認します。監視テーブルに対しての SQL 文実行は行わない簡易的な監視です。監視テーブルに対しての SQL 文実行は行わない簡易的な監視です。

◆ レベル 1(select での監視)

監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（ select ）です。

◆ レベル 2(update/select での監視)

監視テーブルに対して更新も行う監視です。監視テーブルに対して実行する SQL 文は（ update / select ）です。
監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

◆ レベル 3(毎回 create/drop も行う)

監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。監視テーブルに対して実行する SQL 文は（ create / insert / select / drop ）です。

既定値 ： レベル2(update/selectでの監視)

接続文字列(255 バイト以内)

監視するデータベースに対応する接続文字列を設定します。必ず設定してください。

監視方式を「インスタンスのみの監視」とした場合には ORACLE_SID を設定します。

監視方式	ORACLE_HOME	接続文字列	監視レベル
リスナーとインスタンスを監視	入力不要	接続文字列を指定	設定に応じたレベルの監視
リスナーのみ監視	入力した場合、Oracleのコマンドを使用した監視	接続文字列を指定	レベル設定は無視される
	未入力の場合、リスナーを経由したインスタンスへの接続確認	接続文字列を指定	レベル設定は無視される
インスタンスのみ監視	入力した場合、BEQ接続によるインスタンスの確認	ORACLE_SIDを指定する	設定に応じたレベルの監視
	未入力の場合、リスナーを経由したインスタンスの確認となる	接続文字列を指定	設定に応じたレベルの監視

既定値 ： 接続文字列の既定値はなし

ユーザ名(255 バイト以内)

データベースにログインする際のユーザ名を設定します。必ず設定してください。

指定したデータベースにアクセス可能な Oracle ユーザを指定してください。

既定値 ：sys

パスワード(255 バイト以内)

データベースにログインする際のパスワードを設定します。

既定値 ： change_on_install

認証方式

データベースの認証方式を設定します。

既定値 ： SYSDBA

監視テーブル名(255 バイト以内)

データベース上に作成する監視用テーブルの名前を設定します。必ず設定してください。

テーブルの作成・削除を行いますので、運用に使用しているテーブル名と重ならないように注意してください。また、SQL 文の予約語と重ならないようにしてください。
データベースの仕様により監視テーブル名に設定できない文字があります。詳細はデータベースの仕様を確認してください。

既定値 ： orawatch

ORACLE_HOME(255 バイト以内)

ORACLE_HOME に設定しているパス名を指定します。[/] で始まる必要があります。監視方式で「リスナーのみ監視」「インスタンスのみ監視」を選択したときに使用されます。

既定値 : なし

文字コード

Oracle のキャラクタ・セットを設定します。必ず設定してください。

既定値 : JAPANESE_JAPAN.JA16EUC

ライブラリパス(1023 バイト以内)

Oracle Call Interface(OCI)用のライブラリパスを設定します。必ず設定してください。

既定値 : /opt/app/oracle/product/10.2.0/db_1/lib/libclntsh.so.10.1

障害発生時にアプリケーションの詳細情報を採取する

本機能を有効にした場合、Oracle モニタリソースが異常を検出すると、Oracle の詳細情報が採取されます。詳細情報は最大 5 回採取されます。

注: 採取中にクラスタ停止などにより、oracle サービスを停止させた場合、正しい情報が取得できない可能性があります。

既定値 : 無効

採取タイムアウト

詳細情報採取時のタイムアウト値を設定します。

既定値 : 600

Oracle の初期化中またはシャットダウン中をエラーにする

本機能を有効にした場合、Oracle の起動中またはシャットダウン中の状態を検出すると、直ちに監視エラーになります。
Oracle Clusterware 等の連携で Oracle が運用中に自動再起動される場合、本機能を無効にしてください。Oracle の起動中またはシャットダウン中の状態でも監視正常になります。
ただし 1 時間以上 Oracle の起動中またはシャットダウン中の状態が継続すると監視エラーになります。

既定値 : 無効

Oracleモニタリソースの注意事項

動作確認済みの Oracle のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースは、Oracle のインターフェイス(Oracle Call Interface)を利用して、Oracle の監視を行っています。そのため、監視を行うサーバ上に、インターフェイス用のライブラリ(libclntsh.so)がインストールされている必要があります。

仮想マシンリソースで制御する仮想マシンのゲストOS上で動作するOracleデータベースを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 Oracle データベースが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。また、この場合は監視リソースが動作するホスト OS 側に Oracle クライアントをセットアップし、仮想マシン上の Oracle データベースに接続するように接続文字列を設定します。

パラメータで指定した接続文字列・ユーザ名・パスワードなどの値が、監視を行う Oracle の環境と異なる場合、Oracle の監視を行うことができません。各エラー内容を示すメッセージが表示されますので、環境を確認してください。

パラメータのユーザ名に指定するユーザについて、デフォルトでは sys となっていますが、別途監視用ユーザを作成する場合、各監視レベルにおいて以下のアクセス権付与が必要です。（sysdba 権限を与えない場合）

監視レベル	必要な権限
レベル0(データベースステータス)	V$INSTANCEへのSELECT権限
レベル1(selectでの監視)	監視テーブルへのSELECT権限
レベル2(update/selectでの監視)	CREATE TABLE／DROP ANY TABLE／監視テーブルへのINSERT権限／監視テーブルへのUPDATE権限／監視テーブルへのSELECT権限
レベル3(毎回create/dropも行う)	CREATE TABLE／DROP ANY TABLE／監視テーブルへのINSERT権限／監視テーブルへのUPDATE権限／監視テーブルへのSELECT権限

管理者ユーザの認証方式が初期化パラメータ・ファイルで REMOTE_LOGIN_PASSWORDFILE を NONE に設定した OS 認証のみである場合、パラメータのユーザ名には SYSDBA 権限のないデータベースユーザ名を指定して下さい。
SYSDBA 権限のあるデータベースユーザを指定した場合、本モニタリソース起動時にエラーとなり監視を行うことができません。

ユーザ名に sys を指定すると Oracle の監査ログが出力されることがあります。監査ログを大量に出力したくない場合、 sys 以外のユーザ名を指定してください。

データベース作成時のキャラクタ・セットは、OS でサポートされているキャラクタ・セットに合わせてください。本モニタリソースの「文字コード」は、Oracle からのエラーメッセージが発生したときに、CLUSTERPRO の WebManager のアラートビューや OS の messages(syslog) に表示させる言語を選択してください。
ただし、データベース接続時のエラー（ユーザ名不正など）については、上記の対応を行っても正しく表示されないことがあります。
NLS パラメータ、NLS_LANG の設定、詳細な内容については、Oracle 社のマニュアル「グローバリゼーション・サポート・ガイド」を参照してください。

「文字コード」の設定は Oracle 自体の動作には影響を与えません。

CLUSTERPRO が OS の messages(syslog)へ 1 バイト文字以外(ANK 文字以外)をエントリするときには常に EUC コードでエントリします。従ってご使用のディストリビューションによっては messages(syslog)の文字コードが EUC でないため文字化けが発生し正しく表示されないことがあります。

(WebManager のアラートビューについては問題ありません。)

文字化けを発生させたくない場合には「文字コード」の設定に AMERICAN_AMERICA.US7ASCII または AMERICAN_AMERICA.UTF8(ANK 文字を使用する言語)を選択してください。

設定例

・日本語で表示させたい場合

JAPANESE_JAPAN で始まる文字セットを選択してください。

・英語で表示させたい場合

AMERICAN_AMERICA で始まる文字セットを選択してください。

次項の「Oracle モニタリソースの監視方法」で説明する監視レベルについて、「レベル 1」「レベル 2」を選択した場合、手動にて監視テーブルを作成しておく必要があります。
「レベル 1」で監視開始時に監視テーブルが無い場合、監視エラーになります。
「レベル 2」で監視開始時に監視テーブルが無い場合、CLUSTERPRO が監視テーブルを自動で作成します。このとき WebManager のアラートビューに監視テーブルがない旨のメッセージが表示されます。

選択する監視レベル	監視テーブルの事前作成
レベル0(データベースステータス)	必要なし
レベル1(selectでの監視)	必要あり
レベル2(update/selectでの監視)	必要あり
レベル3(毎回create/dropも行う)	必要なし

監視テーブルの作成は以下の手順で行えます。

SQL 文で作成する場合(以下の例は監視テーブル名を orawatch とする場合)
sql> create table orawatch (num number(11,0) primary key);
sql> insert into orawatch values(0);
sql> commit;

※パラメータのユーザ名に指定するユーザのスキーマに作成してください。

CLUSTERPRO のコマンドを利用する場合
clp_oraclew --createtable -n <Oracle モニタリソース名>
※パラメータのユーザ名に指定するユーザが sys 以外で sysdba 権限を付与しないユーザの場合は CREATE TABLE 権限が必要です。
作成した監視テーブルを手動で削除する場合、下記のコマンドを実行してください:
clp_oraclew --deletetable -n <Oracle モニタリソース名>

「レベル 3」での監視は毎回監視テーブルの作成・削除を行うため「レベル 1」「レベル 2」より監視の負荷が高くなります。また Oracle のリソースの使用量が増え続けるため、定期的に Oracle インスタンスを再起動する運用以外での「レベル 3」の監視は推奨しません。

Oracleモニタリソースの監視方法

Oracle モニタリソースは、以下の監視レベルから選択した監視レベルに応じた監視を行います。

- レベル 0(データベースステータス)

  Oracle の管理テーブル（V$INSTANCE 表）を参照し DB の状態（インスタンスの状態）を確認します。監視テーブルに対しての SQL 文実行は行わない簡易的な監視です。

  監視の結果以下の場合に異常とみなします。

  (1) Oracle の管理テーブル(V$INSTANCE 表)のステータス（ status ）が未起動状態（ MOUNTED,STARTED ）の場合

  (2) Oracle の管理テーブル(V$INSTANCE 表)のデータベースステータス（ database_status ）が未起動（ SUSPENDED,INSTANCE RECOVERY ）の場合

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（ select ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ update / select ）です。

  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

- レベル 3(毎回 create/drop も行う)

  監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ create / insert / select / drop ）です。

  監視の結果以下の場合に異常とみなします。

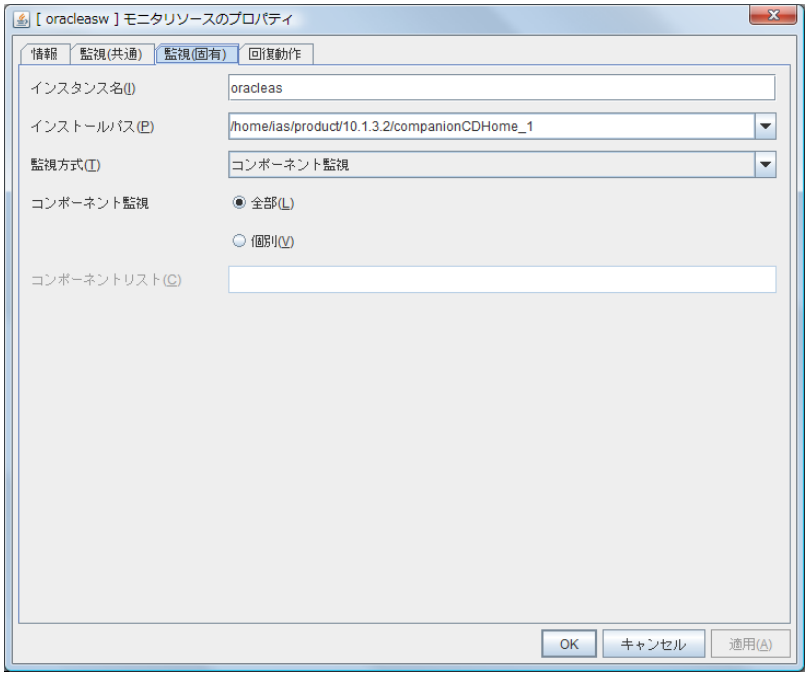
  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

OracleAS モニタリソースの設定

OracleAS モニタリソースは、サーバ上で動作する Oracle アプリケーションサーバを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors]のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の OracleAS モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



インスタンス名(255 バイト以内)

監視するインスタンスを設定します。必ず設定してください。

既定値 : なし

インストールパス(1023 バイト以内)

Oracle アプリケーションのインストールパスを設定します。必ず設定してください。

既定値 : /home/ias/product/10.1.3.2/companionCDHome_1

監視方式

監視対象とする Oracle アプリケーションサーバの機能を選択します。

- opmnプロセスとコンポーネント同時監視

  opmn プロセスの死活監視とコンポーネントのステータス監視を実施します。

- opmnプロセス監視

  opmn プロセスの死活のみ監視します。

- コンポーネント監視(既定値)

  コンポーネントのステータスのみ監視します。

コンポーネント監視

監視方式として[opmn プロセスとコンポーネント同時監視]または[コンポーネント監視]を選択している場合に、監視対象のコンポーネントを個別に指定するかどうかを選択します。

- 全部(既定値)

  全てのコンポーネントを監視します。

- 個別

  コンポーネントリストで指定したコンポーネントのみ監視します。

コンポーネントリスト(1023 バイト以内)

コンポーネント監視の対象コンポーネント名を設定します。複数設定する場合はカンマ“,”で区切ります。[コンポーネント監視]を[個別]に設定している場合は必ず設定してください。

OracleASモニタリソースの注意事項

動作確認済みの Oracle アプリケーションサーバのバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

監視の対象リソースには、Oracle アプリケーションサーバを起動する exec リソースなどを指定してください。対象リソースの活性後、監視を開始しますが、対象リソースの活性直後に Oracle アプリケーションサーバがすぐに動作できない場合などは、[監視開始待ち時間]で調整してください。

監視対象リソースの活性において、Oracle アプリケーションサーバのインスタンスで開始されないコンポーネントが存在する場合、opmn.xml ファイルを編集して該当コンポーネントの status を"disabled"に指定してください。opmn.xml ファイルの詳細については Oracle アプリケーションサーバのマニュアルを参照してください。

監視動作ごとに Oracle アプリケーションサーバ自体が動作ログなどを出力することがありますが、その制御は、Oracle アプリケーションサーバ側の設定で適宜行ってください。

OracleASモニタリソースの監視方法

OracleAS モニタリソースは、以下の監視を行います。

OracleAS の opmnctl コマンドを利用して、アプリケーションサーバの監視を実行します。

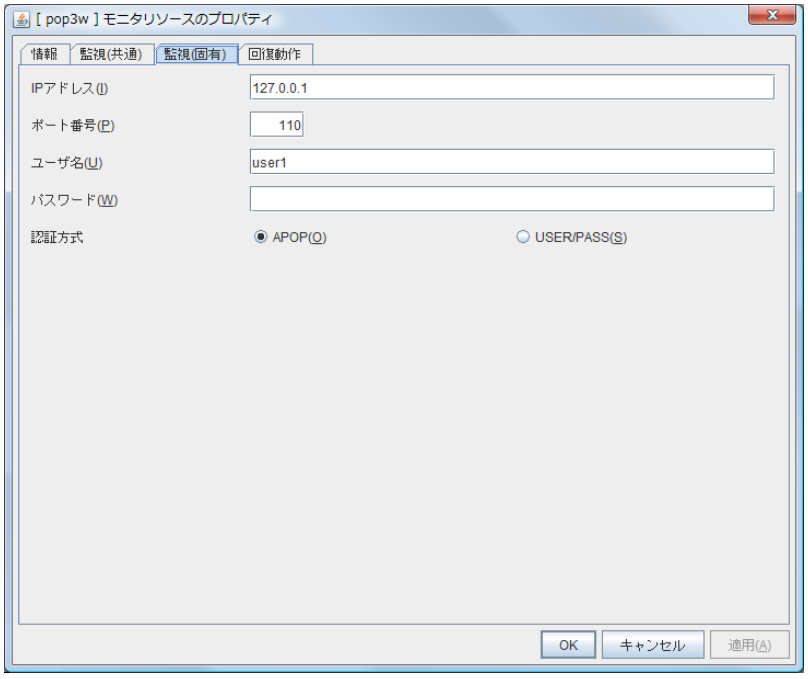
監視の結果、以下の場合に異常とみなします。

(1)取得したアプリケーションサーバの状態で異常が通知された場合

POP3 モニタリソースの設定

POP3 モニタリソースは、サーバ上で動作する POP3 のサービスを監視するモニタリソースです。POP3 プロトコルを監視するものであり、特定のアプリケーションの監視ではありません。そのため、POP3 プロトコルを実装するさまざまなアプリケーションの監視を行うことができます。

1. Builder 左部分に表示されているツリービューで、モニタリソースのアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の POP3 モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



IP アドレス (79 バイト以内)

監視する POP3 サーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する POP3 サーバに接続しますので、ループバックアドレス(127.0.0.1)を設定しますが、POP3 サーバの設定で接続可能なアドレスを制限している場合は、接続可能なアドレス(フローティング IP アドレス等)を設定します。また、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する POP3 サーバを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 : 127.0.0.1

ポート番号 (1～65535)

監視する POP3 のポート番号を設定します。必ず設定してください。

既定値 ： 110

ユーザ名(255 バイト以内)

POP3 にログインする際のユーザ名を設定します。

既定値 ： なし

パスワード(255 バイト以内)

POP3 にログインする際のパスワードを設定します。[変更]ボタンを押してパスワード指定ダイアログを表示して設定します。

既定値 ： なし

認証方式

POP3 にログインするときの認証方式を選択します。使用している POP3 の設定に合わせる必要があります。

- APOP(既定値)

  APOP コマンドを使用した暗号化認証方式です。

- USER/PASS

  USER/PASS コマンドを使用した平文方式です。

POP3モニタリソースの注意事項

監視の対象リソースには、POP3 サーバを起動する exec リソースなどを指定してください。対象リソースの活性後、監視を開始しますが、対象リソースの活性直後に POP3 がすぐに動作できない場合などは、[監視開始待ち時間]で調整してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する POP3 サーバを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 POP3 サーバが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。

監視動作ごとに POP3 自体が動作ログなどを出力することがありますが、その制御は、POP3 側の設定で適宜行ってください。

POP3モニタリソースの監視方法

POP3 モニタリソースは、以下の監視を行います。

POP3 サーバに接続して動作確認コマンドを実行します。

監視の結果、以下の場合に異常とみなします。

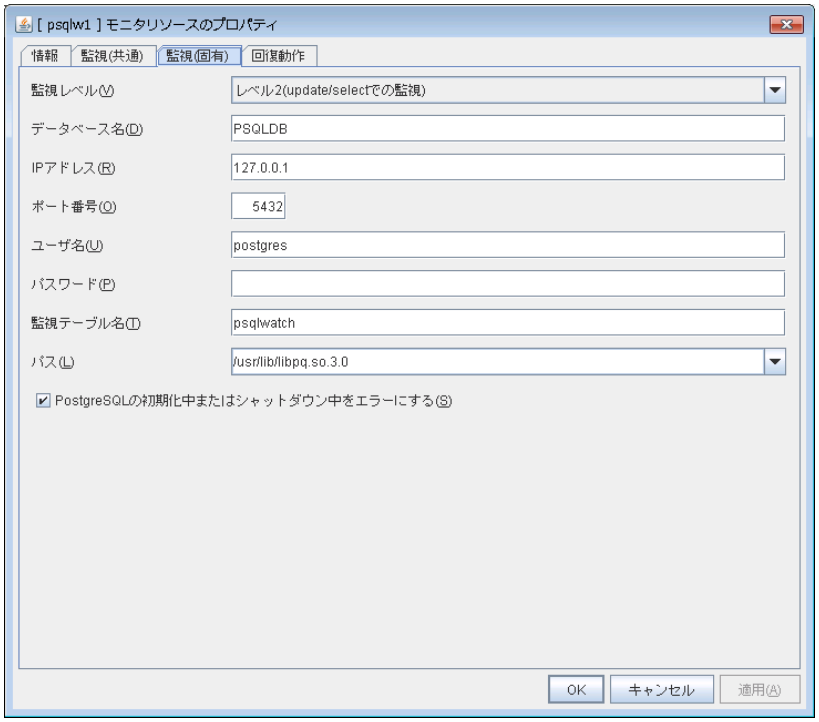
(1) POP3 サーバへの接続に失敗した場合

(2) コマンドに対する応答で異常が通知された場合

PostgreSQL モニタリソースの設定

PostgreSQL モニタリソースは、サーバ上で動作する PostgreSQL のデータベースを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の PostgreSQL モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



監視レベル

選択肢の中から1つを選択します。必ず設定してください。

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（ select ）です。

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。監視テーブルに対して実行する SQL 文は（ update / select / reindex / vacuum ）です。
  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

- レベル 3(毎回 create/drop も行う)

監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。監視テーブルに対して実行する SQL 文は（ create / insert / select / reindex / drop / vacuum ）です。

既定値 ： レベル2(update/selectでの監視)

データベース名(255 バイト以内)

監視するデータベース名を設定します。必ず設定してください。

既定値 ： なし

IP アドレス(79 バイト以内)

接続するサーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する PostgreSQL サーバに接続しますので、ループバックアドレス（127.0.0.1）を設定しますが、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する PostgreSQL データベースを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 ： 127.0.0.1

ポート番号(1～65535)

接続する際のポート番号を設定します。必ず設定してください。

既定値 ： 5432

ユーザ名(255 バイト以内)

データベースにログインする際のユーザ名を設定します。必ず設定してください。

指定したデータベースにアクセス可能な PostgreSQL ユーザを指定してください。

既定値 ： postgres

パスワード(255 バイト以内)

データベースにログインする際のパスワードを設定します。

既定値 ： なし

監視テーブル名(255 バイト以内)

データベース上に作成する監視用テーブルの名前を設定します。必ず設定してください。

テーブルの作成・削除を行いますので、運用に使用しているテーブル名と重ならないように注意してください。また、SQL 文の予約語と重ならないようにしてください。
データベースの仕様により監視テーブル名に設定できない文字があります。詳細はデータベースの仕様を確認してください。

既定値 ： psqlwatch

パス(1023 バイト以内)

PostgreSQL のライブラリパスを設定します。必ず設定してください。

既定値 ： /usr/lib/libpq.so.3.0

PostgreSQL の初期化中またはシャットダウン中をエラーにする

本機能を有効にした場合、PostgreSQL の起動中またはシャットダウン中の状態を検出すると、直ちに監視エラーになります。

本機能を無効にした場合、PostgreSQL の起動中またはシャットダウン中の状態でも監視正常になります。

ただし 1 時間以上 PostgreSQL の起動中またはシャットダウン中の状態が継続すると監視エラーになります。

既定値 ： 有効

PostgreSQLモニタリソースの注意事項

動作確認済みの PostgreSQL のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースは、PostgreSQL の libpq のライブラリを利用して、PostgreSQL の監視を行っています。
本モニタリソースが異常になる場合は、PostgreSQL の libpq ライブラリが存在するパスへアプリケーションのライブラリパスを設定してください。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する PostgreSQL データベースを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 PostgreSQL データベースが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。

パラメータ指定値が、監視を行う PostgreSQL の環境と異なる場合、Web Manager のアラートビューにエラー内容を示すメッセージが表示されますので、環境を確認してください。

クライアント認証について、本モニタリソースでは pg_hba.conf ファイルに設定可能な以下の認証方式が動作確認済みとなっています。
trust、md5、password

本モニタリソースを利用すると PostgreSQL 側のログに下記のようなメッセージが出力されます。監視処理に伴って出力されるメッセージで、問題はありません。
YYYY-MM-DD hh:mm:ss JST moodle moodle LOG: statement: DROP TABLE psqlwatch
YYYY-MM-DD hh:mm:ss JST moodle moodle ERROR: table "psqlwatch" does not exist
YYYY-MM-DD hh:mm:ss JST moodle moodle STATEMENT: DROP TABLE psqlwatch
YYYY-MM-DD hh:mm:ss JST moodle moodle LOG: statement: CREATE TABLE psqlwatch (num INTEGER NOT NULL PRIMARY KEY)
YYYY-MM-DD hh:mm:ss JST moodle moodle NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "psqlwatch_pkey" for table "psql watch"
YYYY-MM-DD hh:mm:ss JST moodle moodle LOG: statement: DROP TABLE psqlwatch

次項の「PostgreSQL モニタリソースの監視方法」で説明する監視レベルについて、「レベル 1」「レベル 2」を選択した場合、手動にて監視テーブルを作成しておく必要があります。
「レベル 1」で監視開始時に監視テーブルが無い場合、監視エラーになります。
「レベル 2」で監視開始時に監視テーブルが無い場合、CLUSTERPRO が監視テーブルを自動で作成します。このとき WebManager のアラートビューに監視テーブルがない旨のメッセージが表示されます。
「レベル 3」での監視は毎回監視テーブルの作成・削除を行うため「レベル 1」「レベル 2」より監視の負荷が高くなります。

選択する監視レベル	監視テーブルの事前作成
レベル1(selectでの監視)	必要あり
レベル2(update/selectでの監視)	必要あり
レベル3(毎回create/dropも行う)	必要なし

監視テーブルの作成は以下の手順で行えます。

SQL 文で作成する場合（以下の例は監視テーブル名を psqlwatch とする場合）
sql> CREATE TABLE psqlwatch (num INTEGER NOT NULL PRIMARY KEY);
sql> INSERT INTO psqlwatch VALUES(0) ;
sql> COMMIT;

CLUSTERPRO のコマンドを利用する場合
clp_psqlw --createtable -n <PostgreSQL モニタリソース名>
作成した監視テーブルを手動で削除する場合、下記のコマンドを実行してください:
clp_psqlw --deletetable -n <PostgreSQL モニタリソース名>

PostgreSQLモニタリソースの監視方法

PostgreSQL モニタリソースは、以下の監視レベルから選択した監視レベルに応じた監視を行います。

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（ select ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ update / select / reindex / vacuum ）です。

  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

- レベル 3(毎回 create/drop も行う)

  監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は（ create / insert / select / reindex / drop / vacuum ）です。

  監視の結果以下の場合に異常とみなします。

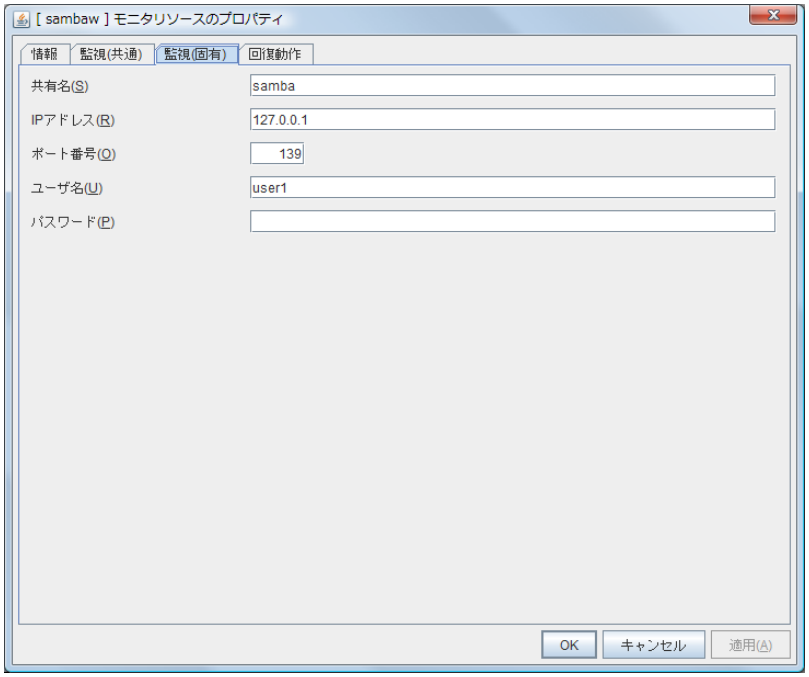
  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

Samba モニタリソースの設定

Samba モニタリソースは、サーバ上で動作する samba のファイルサーバを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の samba モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



共有名(255 バイト以内)

監視を行う samba サーバの共有名を設定します。必ず設定してください。

既定値 : なし

IP アドレス(79 バイト以内)

samba サーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する samba のファイルサーバに接続しますので、ループバックアドレス(127.0.0.1)を設定しますが、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する samba のファイルサーバを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 : 127.0.0.1

ポート番号(1～65535)

samba デーモンが使用しているポート番号を設定します。必ず設定してください。

既定値 ： 139

ユーザ名(255 バイト以内)

samba サービスにログインする際のユーザ名を設定します。必ず設定してください。

既定値 ： なし

パスワード(255 バイト以内)

samba サービスにログインする際のパスワードを設定します。

既定値 ： なし

Sambaモニタリソースの注意事項

動作確認済みの samba のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースが異常になる場合は、パラメータの設定値とsambaの環境が一致していない可能性がありますので、環境を確認してください。

監視する共有名については、自サーバから接続できるように smb.conf を設定してください。また、smb.conf ファイルの security パラメータが share の場合は、ゲスト接続を有効にしてください。

ファイル共有、プリンタ共有以外の samba の機能に関しては監視を行いません。

仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する samba のファイルサーバを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 samba のファイルサーバが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。

samba の認証モードが Domain もしくは Server の場合、監視サーバ上で smbmount を実行すると、本モニタリソースのパラメータで指定したユーザ名で mount されることがあります。

Sambaモニタリソースの監視方法

Samba モニタリソースは、以下の監視を行います。

samba サーバに接続して samba サーバのリソースに対する tree connection の確立を確認します。

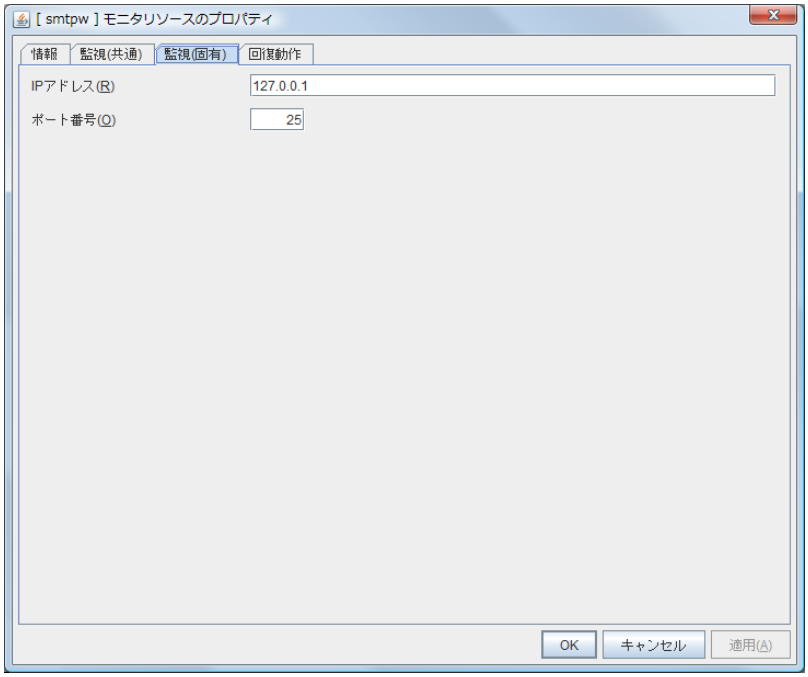
監視の結果、以下の場合に異常とみなします。

(1)samba サービスへの要求に対する応答内容が不正な場合

SMTP モニタリソースの設定

SMTP モニタリソースは、サーバ上で動作する SMTP デーモンを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の SMTP モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



IP アドレス(79 バイト以内)

監視する SMTP サーバの IP アドレスを設定します。必ず設定してください。

通常は自サーバ上で動作する SMTP サーバに接続しますので、ループバックアドレス(127.0.0.1)を設定しますが、仮想マシンリソースで制御する仮想マシンのゲスト OS 上で動作する SMTP サーバを監視する場合は、仮想マシンの IP アドレスを設定します。

既定値 : 127.0.0.1

ポート番号(1～65535)

SMTP サーバに接続する際のポート番号を設定します。必ず設定してください。

既定値 : 25

SMTPモニタリソースの注意事項

動作確認済みの SMTP のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

ロードアベレージが、sendmail.def ファイルで設定されている RefuseLA の値よりも超えた状態が一定時間続くと、本モニタリソースで異常とみなすことがあります。

仮想マシンリソースで制御する仮想マシンのゲストOS上で動作するSMTPサーバを監視する場合は、監視の対象リソースとして仮想マシンリソースを指定し、仮想マシンリソース活性後 SMTP サーバが接続可能となるまでの十分な待ち時間を[監視開始待ち時間] に設定してください。

SMTPモニタリソースの監視方法

SMTP モニタリソースは、以下の監視を行います。

サーバ上のSMTPデーモンに接続し、NOOPコマンドの発行により、SMTPデーモンの監視を実行します。

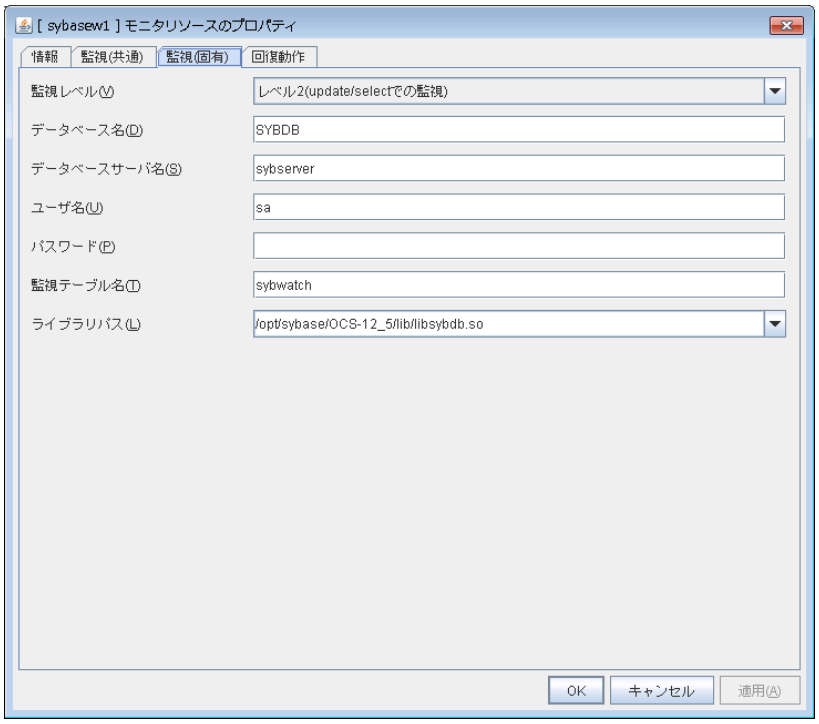
監視の結果、以下の場合に異常とみなします。

(1)SMTP デーモンへの接続や NOOP コマンドの発行に対する応答で異常が通知された場合

Sybase モニタリソースの設定

Sybase モニタリソースは、サーバ上で動作する Sybase のデータベースを監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の Sybase モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



監視レベル

選択肢の中から1つを選択します。必ず設定してください。

- レベル 0(データベースステータス)

  Sybase の管理テーブル（ sys.sysdatabases ）を参照し DB の状態を確認します。監視テーブルに対して SQL 文の発行は行わない簡易的な監視です。

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は（select ）です。

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。監視テーブルに対して実行する SQL 文は（ update / select ）です。

監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は（ create / insert ）です。

- レベル 3(毎回 create/drop も行う)

  監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。監視テーブルに対して実行する SQL 文は（ create / insert / select / drop ）です。

- 既定値 ： レベル 2(update/select での監視)

データベース名(255 バイト以内)

監視するデータベース名を設定します。必ず設定してください。

既定値 ： なし

データベースサーバ名(255 バイト以内)

監視するデータベースサーバ名を設定します。必ず設定してください。

既定値 ： なし

ユーザ名(255 バイト以内)

データベースにログインする際のユーザ名を設定します。必ず設定してください。

指定したデータベースにアクセス可能な Sybase ユーザを指定してください。

既定値 ： sa

パスワード(255 バイト以内)

データベースにログインする際のパスワードを設定します。

既定値 ： なし

監視テーブル名(255 バイト以内)

データベース上に作成する監視用テーブルの名前を設定します。必ず設定してください。

テーブルの作成・削除を行いますので、運用に使用しているテーブル名と重ならないように注意してください。また、SQL 文の予約語と重ならないようにしてください。
データベースの仕様により監視テーブル名に設定できない文字があります。詳細はデータベースの仕様を確認してください。

既定値 ： sybwatch

ライブラリパス(1023 バイト以内)

Sybase のライブラリパスを設定します。必ず設定してください。

既定値 ： /opt/sybase/OCS-12_5/lib/libsybdb.so

Sybaseモニタリソースの注意事項

動作確認済みの Sybase のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースは、ASE の Open Client DB-Library/C を使用して、ASE の監視を行っています。

パラメータ指定値が、監視を行う ASE の環境と異なる場合、WebManager のアラートビューにエラー内容を示すメッセージが表示されますので、環境を確認してください。

次項の「Sybase モニタリソースの監視方法」で説明する監視レベルについて、「レベル 1」「レベル 2」を選択した場合、手動にて監視テーブルを作成しておく必要があります。
「レベル 1」で監視開始時に監視テーブルが無い場合、監視エラーになります。
「レベル 2」で監視開始時に監視テーブルが無い場合、CLUSTERPRO が監視テーブルを自動で作成します。このとき WebManager のアラートビューに監視テーブルがない旨のメッセージが表示されます。
「レベル 3」での監視は毎回監視テーブルの作成・削除を行うため「レベル 1」「レベル 2」より監視の負荷が高くなります。

選択する監視レベル	監視テーブルの事前作成
レベル0(データベースステータス)	必要なし
レベル1(selectでの監視)	必要あり
レベル2(update/selectでの監視)	必要あり
レベル3(毎回create/dropも行う)	必要なし

監視テーブルの作成は以下の手順で行えます。

SQL 文で作成する場合（以下の例は監視テーブル名を sybwatch とする場合）
sql> CREATE TABLE sybwatch (num INT NOT NULL PRIMARY KEY)
sql> GO
sql> INSERT INTO sybwatch VALUES(0)
sql> GO
sql> COMMIT
sql> GO

CLUSTERPRO のコマンドを利用する場合
clp_sybasew --createtable -n <Sybase モニタリソース名>
作成した監視テーブルを手動で削除する場合、下記のコマンドを実行してください:
clp_sybasew --deletetable -n <Sybase モニタリソース名>

Sybaseモニタリソースの監視方法

Sybase モニタリソースは、以下の監視レベルから選択した監視レベルに応じた監視を行います。

◆ レベル 0(データベースステータス)

Sybase の管理テーブル（ sys.sysdatabases ）を参照し DB の状態を確認します。監視テーブルに対して SQL 文の発行は行わない簡易的な監視です。

監視の結果以下の場合に異常とみなします。

(1) データベースのステータスがオフラインといった使用不可状態の場合

- レベル 1(select での監視)

  監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行する SQL 文は(select)です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

- レベル 2(update/select での監視)

  監視テーブルに対して更新も行う監視です。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は(update / select)です。

  監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行する SQL 文は(create / insert)です。

  監視の結果以下の場合に異常とみなします。

  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

- レベル 3(毎回 create/drop も行う)

  監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。SQL 文の実行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行する SQL 文は(create / insert / select / drop)です。

  監視の結果以下の場合に異常とみなします。

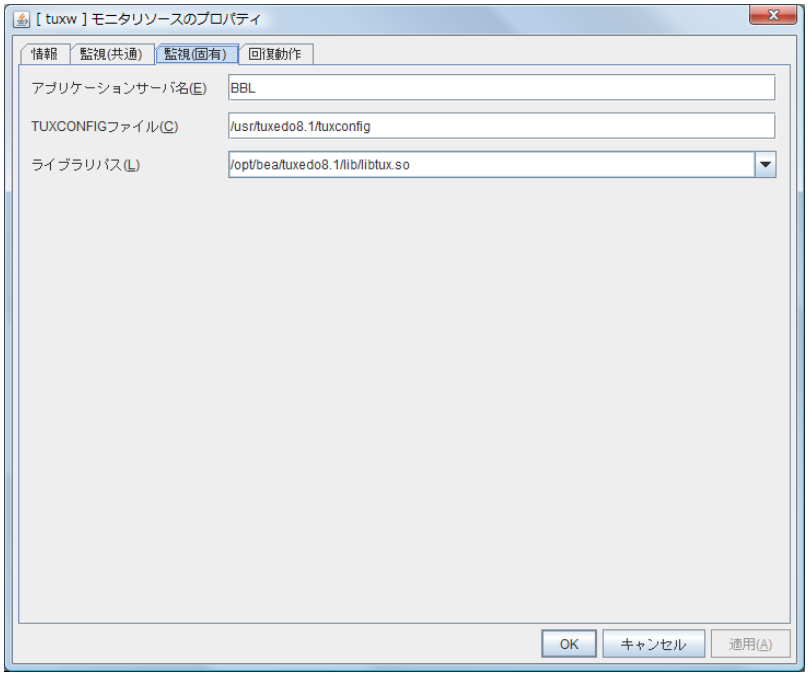
  (1) データベースへの接続や SQL 文の発行に対する応答で異常が通知された場合

  (2) 書き込んだデータと読み込んだデータが一致していない場合

Tuxedo モニタリソースの設定

Tuxedo モニタリソースは、サーバ上で動作する Tuxedo を監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の Tuxedo モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



アプリケーションサーバ名(255 バイト以内)

監視するアプリケーションサーバ名を設定します。必ず設定してください。

既定値 : BBL

TUXCONFIG ファイル名(1023 バイト以内)

Tuxedo の配置ファイル名を設定します。必ず設定してください。

既定値 : なし

ライブラリパス(1023 バイト以内)

Tuxedo のライブラリパスを設定します。必ず設定してください。

既定値 : /opt/bea/tuxedo8.1/lib/libtux.so

Tuxedoモニタリソースの注意事項

動作確認済みの Tuxedo のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

Tuxedo のライブラリ(libtux.so など)が存在しない場合、監視を行うことができません。

Tuxedoモニタリソースの監視方法

Tuxedo モニタリソースは、以下の監視を行います。

Tuxedo の API を利用して、アプリケーションサーバの監視を実行します。

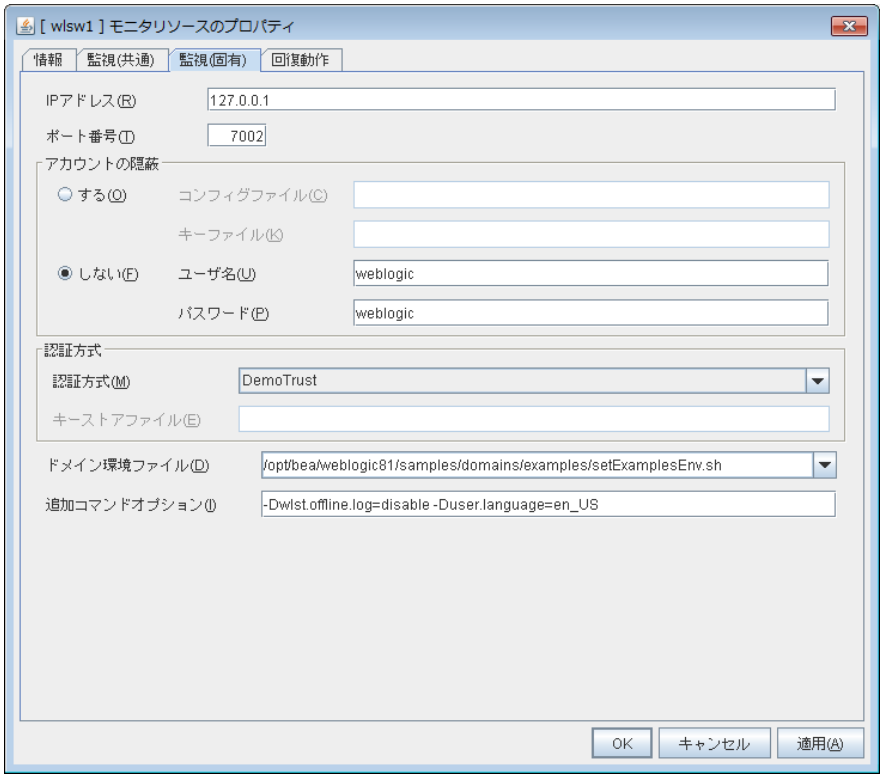
監視の結果、以下の場合に異常とみなします。

(1)アプリケーションサーバへの接続や状態取得に対する応答で異常が通知された場合

Weblogic モニタリソースの設定

Weblogic モニタリソースは、サーバ上で動作する Weblogic を監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の Weblogic モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



IP アドレス(79 バイト以内)

監視するサーバの IP アドレスを設定します。必ず設定してください。

既定値 ： 127.0.0.1

ポート番号(1～65535)

サーバに接続する際のポート番号を設定します。必ず設定してください。

既定値 ： 7002

アカウントの隠蔽

ユーザ名とパスワードを直接指定する場合は「しない」を、ファイル内に記述する場合は「する」を指定してください。必ず設定してください。

既定値 ： しない

コンフィグファイル (1023 バイト以内)

ユーザ情報を保持しているファイル名を設定します。アカウントの隠蔽「する」の場合、必ず設定してください。

既定値 ： なし

キーファイル名(1023 バイト以内)

コンフィグファイルパスにアクセスするためのパスワードを保存しているファイル名を、フルパスで設定します。アカウントの隠蔽「する」の場合、必ず設定してください。

既定値 ： なし

ユーザ名(255 バイト以内)

Weblogic のユーザ名を設定します。アカウントの隠蔽「しない」の場合、必ず設定してください。

既定値 ：weblogic

パスワード(255 バイト以内)

Weblogic のパスワードを設定します。

既定値 ： weblogic

認証方式

アプリケーションサーバに接続する際の認証方式を設定します。必ず設定してください。

既定値 ： DemoTrust

キーストアファイル(1023 バイト以内)

SSL 認証時の認証ファイルを設定します。認証方式が「CustomTrust」の場合、必ず設定してください。

既定値 ： なし

ドメイン環境ファイル(1023 バイト以内)

Weblogic のドメイン環境ファイル名を設定します。必ず設定してください。

既定値 ： /opt/bea/weblogic81/samples/domains/examples/setExamplesEnv.sh

追加コマンドオプション (1023 バイト以内)

[webLogic.WLST] コマンドへ渡すオプションを変更する場合に設定します。

既定値 : -Dwlst.offline.log=disable -Duser.language=en_US

Weblogicモニタリソースの注意事項

動作確認済みの Weblogic のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースで監視を行うために JAVA 環境が必要です。アプリケーションサーバシステムは JAVA の機能を利用しているため、JAVA のストールなどが発生した場合も異常とみなすことがあります。

WebLogic 起動時にすぐに動作できない場合は異常とみなしてしまうため、[監視開始待ち時間] で調整してください。もしくは、WebLogic を先に起動するようにしてください(例:監視の対象リソースに、WebLogic を起動するスクリプトリソースを指定)。

Weblogicモニタリソースの監視方法

Weblogic モニタリソースは、以下の監視を行います。

[webLogic.WLST] コマンドを利用して connect を行うことで、アプリケーションサーバの監視を実行します。

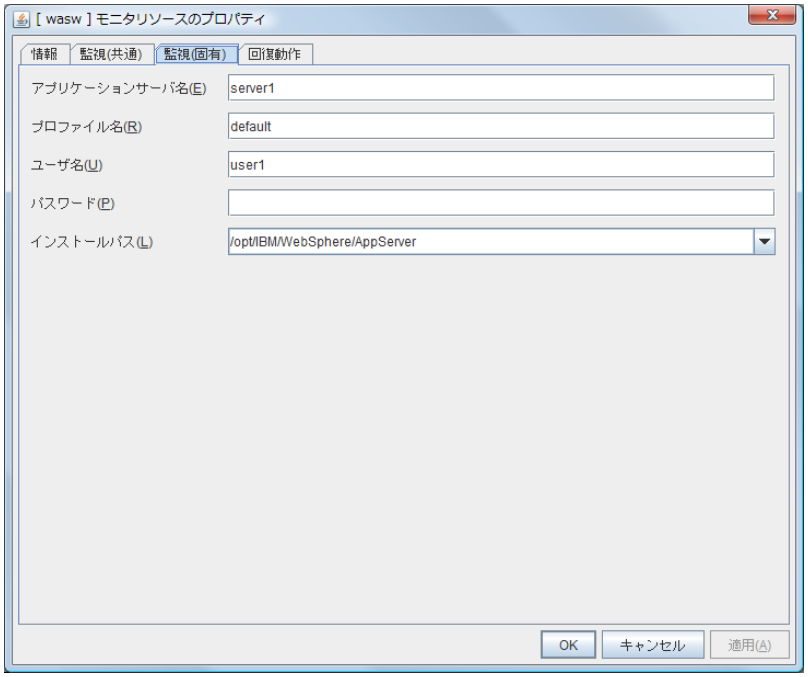
監視の結果、以下の場合に異常とみなします。

(1) connect の応答で異常が通知された場合

Websphere モニタリソースの設定

Websphere モニタリソースは、サーバ上で動作する Websphere を監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の Websphere モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



アプリケーションサーバ(255 バイト以内)

監視するアプリケーションサーバ名を設定します。必ず設定してください。

既定値 : server1

プロファイル名(1023 バイト以内)

監視するアプリケーションサーバのプロファイル名を設定します。必ず設定してください。

既定値 : default

ユーザ名(255 バイト以内)

Websphere のユーザ名を設定します。必ず設定してください。

既定値 :なし

パスワード(255 バイト以内)

Websphere のパスワードを設定します。

既定値 ： なし

インストールパス(1023 バイト以内)

Websphere のインストールパスを設定します。必ず設定してください。

既定値 ： /opt/IBM/WebSphere/AppServer

Websphereモニタリソースの注意事項

動作確認済みの Websphere のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースで監視を行うためには JAVA 環境が必要です。アプリケーションサーバシステムは JAVA の機能を利用しているため、JAVA のストールなどが発生した場合も異常とみなすことがあります。

Websphereモニタリソースの監視方法

Websphere モニタリソースは、以下の監視を行います。

Websphere の serverStatus.sh コマンドを利用して、アプリケーションサーバの監視を実行します。

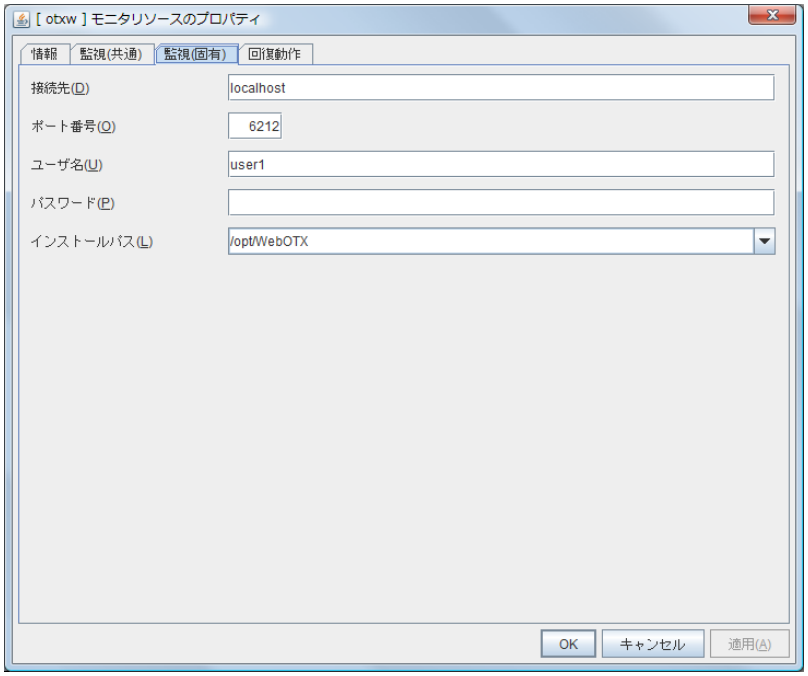
監視の結果、以下の場合に異常とみなします。

(1)取得したアプリケーションサーバの状態で異常が通知された場合

WebOTX モニタリソースの設定

WebOTX モニタリソースは、サーバ上で動作する WebOTX を監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の WebOTX モニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



接続先(255 バイト以内)

監視するサーバのサーバ名を設定します。必ず設定してください。

既定値 : localhost

ポート番号(1～65535)

サーバに接続する際のポート番号を設定します。必ず設定してください。

WebOTX ユーザドメインを監視する場合、WebOTX ドメインの管理ポート番号を設定してください。管理ポート番号とは、ドメイン作成時に <ドメイン名>.properties の domain.admin.port にて設定したポート番号です。<ドメイン名>.properties の詳細については WebOTX のドキュメントを参照してください。

既定値 : 6212

ユーザ名(255 バイト以内)

WebOTX のユーザ名を設定します。必ず設定してください。

WebOTX ユーザドメインを監視する場合、WebOTX ドメインのログインユーザ名を設定してください。

既定値 :なし

パスワード(255 バイト以内)

WebOTX のパスワードを設定します。

既定値 ： なし

インストールパス(1023 バイト以内)

WebOTX のインストールパスを設定します。必ず設定してください。

既定値 ： /opt/WebOTX

WebOTXモニタリソースの注意事項

動作確認済みの WebOTX のバージョンについては、『インストールガイド』の『監視オプションの動作確認済アプリケーション情報』を参照してください。

本モニタリソースで監視を行うためには JAVA 環境が必要です。アプリケーションサーバシステムは JAVA の機能を利用しているため、JAVA のストールなどが発生した場合も異常とみなすことがあります。

WebOTXモニタリソースの監視方法

WebOTX モニタリソースは、以下の監視を行います。

WebOTX の otxadmin.sh コマンドを利用して、アプリケーションサーバの監視を実行します。

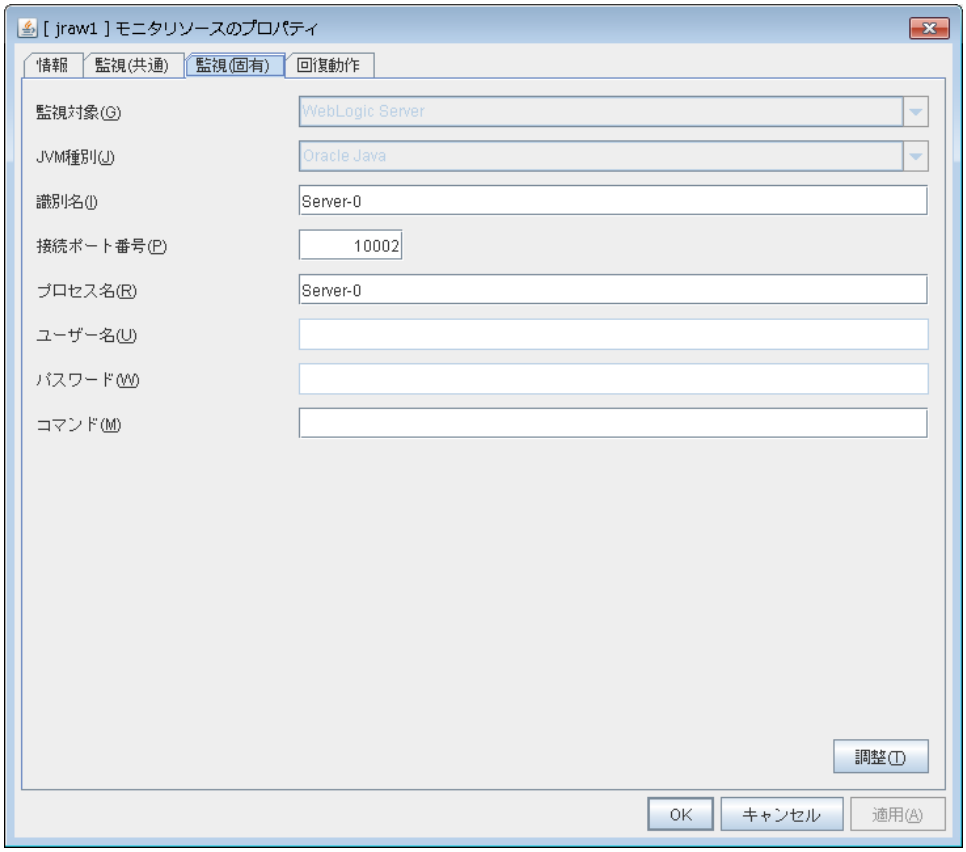
監視の結果、以下の場合に異常とみなします。

(1)取得したアプリケーションサーバの状態で異常が通知された場合

JVM モニタリソースの設定

JVM モニタリソースは、サーバ上で動作する Java VM やアプリケーションサーバが使用するリソースの利用情報を監視するモニタリソースです。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的の JVM モニタリソース名を右クリックし、[プロパティ] の [パラメータ] タブをクリックします。
3. [パラメータ] タブで、以下の説明に従い詳細設定の表示/変更を行います。



監視対象

監視対象をリストから選択します。WebSAM SVF for PDF、WebSAM Report Director Enterprise、WevSAM Universal Connect/X を監視する場合は、[WebSAM SVF]を選択してください。自製の Java アプリケーションを監視する場合は、[Java アプリケーション]を選択してください。

JBoss Enterprise Application Platform 5 以前および JBoss Enterprise Application Platform 6 以降のスタンドアローンモードを監視する場合は[JBoss]、JBoss Enterprise Application Platform 6 以降のドメインモードを監視する場合は「JBoss ドメインモード」を選択してください。

既定値 : なし

JVM 種別

監視対象のアプリケーションが動作する Java VM をリストから選択します。

Java 8 以降の場合は、[Oracle Java(usage monitoring)]を選択してください。Java 8 では以下の仕様変更がありました。

- 非ヒープ領域における各メモリの最大値が取得できなくなりました。
- Perm Gen は Metaspace に変更されました。

そのため、Java 8では[メモリ]タブの監視項目は以下に変更となります。

- 使用率監視は使用量監視に変更となります。
- [Perm Gen]、Perm Gen[shared-ro]、Perm Gen[shared-rw]は監視できません。チェックボックスはオフにしてください。
- [Metaspace]を監視可能です。

Java 9 では以下の仕様変更がありました。

- Code Cache が分割されました。

そのため、Java 9 では[メモリ]タブの監視項目は以下に変更となります。

- [Code Cache]は監視できません。チェックボックスはオフにしてください。

監視対象別では以下を指定可能です。

- 監視対象が[WebLogic Server]の場合
  [Oracle Java]、[Oracle Java(usage monitoring)]、[Oracle JRockit] が選択可能です。
- 監視対象が[Tomcat]の場合
  [Oracle Java]、[Oracle Java(usage monitoring)]、[OpenJDK] が選択可能です。
- 監視対象が[WebLogic Server] [Tomcat]以外の場合
  [Oracle Java]、[Oracle Java(usage monitoring)] が選択可能です。

既定値 : なし

識別名(255 バイト以内)

識別名とは、JVM モニタリソースの JVM 運用ログに監視対象の情報を出力する際に、別の JVM モニタリソースと識別するために設定します。そのため、JVM モニタリソース間で一意の文字列を設定してください。必ず設定してください。

- 監視対象が「WebLogic Server」の場合
  「WebLogic Serverを監視するには」の2を参照して、監視対象のサーバインスタンス名を設定してください。
- 監視対象が「WebOTX プロセスグループ」の場合
  プロセスグループ名を設定してください。
- 監視対象が「WebOTX ドメインエージェント」の場合
  ドメイン名を設定してください。
- 監視対象が「JBoss」「JBoss ドメインモード」の場合
  「JBossを監視するには」を参照して設定してください。

- 監視対象が「Tomcat」の場合
  「Tomcatを監視するには」を参照して設定してください。
- 監視対象が「WebOTX ESB」の場合
  「WebOTX プロセスグループ」の場合と同じです。
- 監視対象が「WebSAM SVF」の場合
  「SVFを監視するには」を参照して設定してください。
- 監視対象が「iPlanet Web Server」の場合
  「iPlanet Web Serverを監視するには」を参照して設定してください。
- 監視対象が「Javaアプリケーション」の場合
  監視対象のJava VM プロセスを一意に識別可能な文字列を指定してください。

既定値 : なし

接続ポート番号(1024～65535)

JVM モニタリソースが、監視対象 Java VM と JMX 接続を行う際に使用するポート番号を設定します。JVM モニタリソースは監視対象 Java VM に JMX 接続を行うことにより情報を取得します。そのため JVM モニタリソースを登録する場合は、監視対象 Java VM に JMX 接続用ポートを開放する設定を行う必要があります。必ず設定してください。クラスタ内のサーバにおいて、共通の設定となります。42424～61000 は推奨しません。

- 監視対象が「WebLogic Server」の場合
  接続ポート番号は「WebLogic Serverを監視するには」の6を参照して設定してください。
- 監視対象が「WebOTX プロセスグループ」の場合
  「WebOTX プロセスグループのJava プロセスを監視するには」を参照して設定してください。
- 監視対象が「WebOTX ドメインエージェント」の場合
  ”(WebOTXインストールパス)/<*ドメイン名*>.properties”の”domain.admin.port”を設定してください。
- 監視対象が「JBoss」の場合
  「JBossを監視するには」を参照して設定してください。
- 監視対象が「JBoss ドメインモード」の場合
  設定不要です。
- 監視対象が「Tomcat」の場合
  「Tomcatを監視するには」を参照して設定してください。
- 監視対象が「WebOTX ESB」の場合
  「WebOTX プロセスグループ」の場合と同じです。
- 監視対象が「WebSAM SVF」の場合
  「SVFを監視するには」を参照して設定してください。
- 監視対象が「iPlanet Web Server」の場合
  「iPlanet Web Serverを監視するには」を参照して設定してください。
- 監視対象が「Javaアプリケーション」の場合
  接続ポート番号は監視対象であるJavaアプリケーションに確認の上、設定してください。

既定値 : なし

プロセス名(1024 バイト以内)

プロセス名とは、JVM モニタリソースの JVM 運用ログに監視対象の情報を出力する際に、別の JVM モニタリソースと識別するために設定します。そのため、JVM モニタリソース間で一意の文字列を設定してください。

- 監視対象が「WebLogic Server」の場合
  監視対象のJava VM プロセスを一意に識別可能な文字列として、監視対象のサーバインスタンス名を指定してください。サーバインスタンス名のみで一意にならない場合は、接続ポート番号と組み合わせてください。
  例)サーバ名がServerA、接続ポート番号が7001の場合: プロセス名はServerA7001。
- 監視対象が「WebOTX プロセスグループ」の場合
  プロセスグループ名を設定してください。多重度設定をしている場合、同じプロセスグループ名が複数指定されないように、同プロセスグループ間で一意に識別可能な文字列を指定してください。
- 監視対象が「WebOTX ドメインエージェント」の場合
  "-Dwebotx.funcid=agent -Ddomain.name=<*ドメイン名*>"を設定してください。
- 監視対象が「JBoss」の場合
  「JBossを監視するには」を参照して設定してください。
- 監視対象が「Tomcat」の場合
  「Tomcatを監視するには」を参照して設定してください。
- 監視対象が「WebOTX ESB」の場合
  「WebOTX プロセスグループ」の場合と同じです。
- 監視対象が「WebSAM SVF」の場合
  「SVFを監視するには」を参照して設定してください。
- 監視対象が「iPlanet Web Server」の場合
  「iPlanet Web Serverを監視するには」を参照して設定してください。
- 監視対象が「Javaアプリケーション」の場合
  監視対象のJava VM プロセスを一意に識別可能な文字列を指定してください。

既定値 : なし

ユーザ名(255 バイト以内)

監視対象の Java VM に接続する管理ユーザ名を設定します。

- 監視対象に[WebOTXドメインエージェント]を選択した場合
  ”/opt/WebOTX/<*ドメイン名*>.properties”の”domain.admin.user”の値を設定してください。
- 監視対象が[WebOTXドメインエージェント]以外の場合
  設定できません。

既定値 :なし

パスワード(255 バイト以内)

監視対象の Java VM に接続する管理ユーザのパスワードを設定します。

- 監視対象に[WebOTXドメインエージェント]を選択した場合
  ”/opt/WebOTX/<*ドメイン名*>.properties”の”domain.admin.passwd”の値を設定してください。
- 監視対象が[WebOTXドメインエージェント]以外の場合
  設定できません。

既定値 ： なし

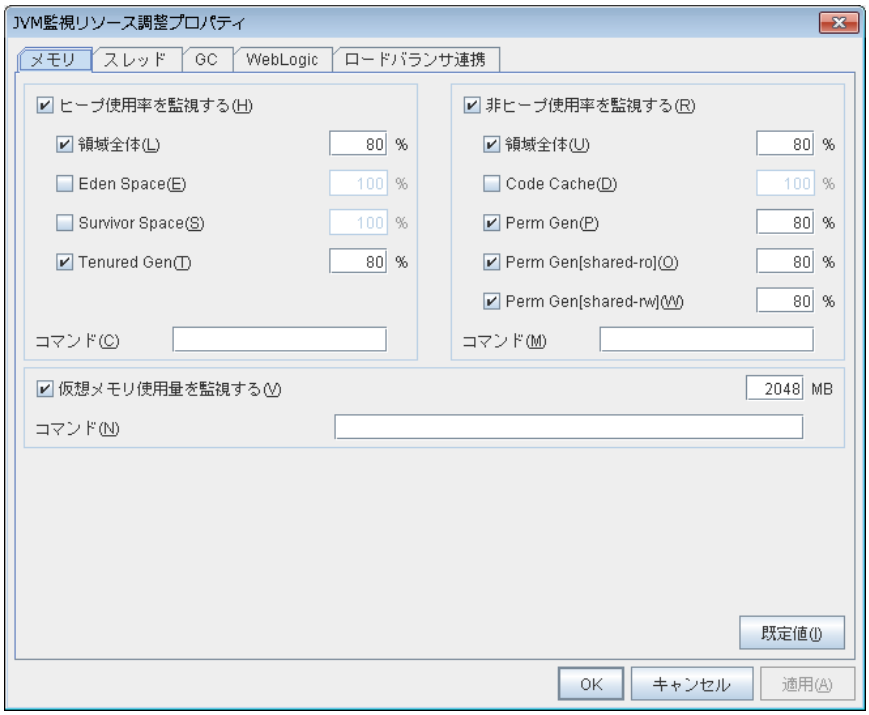
コマンド(255 バイト以内)

監視対象の Java VM 異常検出時に、実行するコマンドを設定します。異常の原因別に実行するコマンドおよび引数の指定が可能です。絶対パスで指定してください。また、実行ファイル名は二重引用符(“”)で括ってください。例)”/usr/local/bin/command” arg1 arg2
ここでは監視対象 Java VM に接続できない場合や使用リソース量の取得における異常検出時に、実行するコマンドを設定します。
「異常検出時に障害原因別にコマンドを実行するには」も参照してください。

既定値 ： なし

さらに[調整]ボタンを選択すると以下の内容がポップアップダイアログに表示されます。以下の説明に従い詳細設定を行います。

メモリタブ（[JVM種別]で[Oracle Java]、[OpenJDK] 選択時）



ヒープ使用率を監視する

監視対象の Java VM が使用する Java ヒープ領域の使用率の監視設定をします。

- チェックボックスがオン(既定値)
  監視します。
- チェックボックスがオフ
  監視しません。

領域全体(1～100)

監視対象の Java VM が使用する Java ヒープ領域の使用率のしきい値を設定します。
既定値 ： 80[%]

Eden Space(1～100)

監視対象の Java VM が使用する Java Eden Space の使用率のしきい値を設定します。GC 方式として G1 GC を指定している場合、G1 Eden Space と読み替えてください。
既定値 ： 100[%]

Survivor Space(1～100)

監視対象の Java VM が使用する Java Survivor Space の使用率のしきい値を設定します。GC 方式として G1 GC を指定している場合、G1 Survivor Space と読み替えてください。
既定値 ： 100[%]

Tenured Gen(1～100)

監視対象の Java VM が使用する Java Tenured(Old) Gen 領域の使用率のしきい値を設定します。GC 方式として G1 GC を指定している場合、G1 Old Gen と読み替えてください。

既定値 : 80[%]

非ヒープ使用率を監視する

監視対象の Java VM が使用する Java 非ヒープ領域の使用率の監視設定をします。

- チェックボックスがオン(既定値)
  監視します。
- チェックボックスがオフ
  監視しません。

領域全体(1～100)

監視対象の Java VM が使用する Java 非ヒープ領域の使用率のしきい値を設定します。

既定値 : 80[%]

Code Cache(1～100)

監視対象の Java VM が使用する Java Code Cache 領域の使用率のしきい値を設定します。

既定値 : 100[%]

Perm Gen(1～100)

監視対象の Java VM が使用する Java Perm Gen 領域の使用率のしきい値を設定します。

既定値 : 80[%]

Perm Gen[shared-ro](1～100)

監視対象の Java VM が使用する Java Perm Gen [shared-ro]領域の使用率のしきい値を設定します。

Java Perm Gen [shared-ro]領域は監視対象 Java VM の起動オプションに -client -Xshare:on -XX:+UseSerialGC を付与して起動している場合に使用される領域です。

既定値 : 80[%]

Perm Gen[shared-rw](1～100)

監視対象の Java VM が使用する Java Perm Gen [shared-rw]領域の使用率のしきい値を設定します。

Java Perm Gen [shared-rw]領域は監視対象 Java VM の起動オプションに -client -Xshare:on -XX:+UseSerialGC を付与して起動している場合に使用される領域です。

既定値 : 80[%]

仮想メモリ使用量を監視する(1～3072)

監視対象の Java VM が使用する仮想メモリ使用量のしきい値を設定します。JVM モニタリソースでは、監視対象の Java VM が 64bit プロセスの場合は仮想メモリ使用量の監視は対応しておりません。そのため、監視対象の Java VM が 64bit プロセスの場合はチェックボックスをオフにしてください。

既定値 : 2048[MB]

コマンド(255 バイト以内)

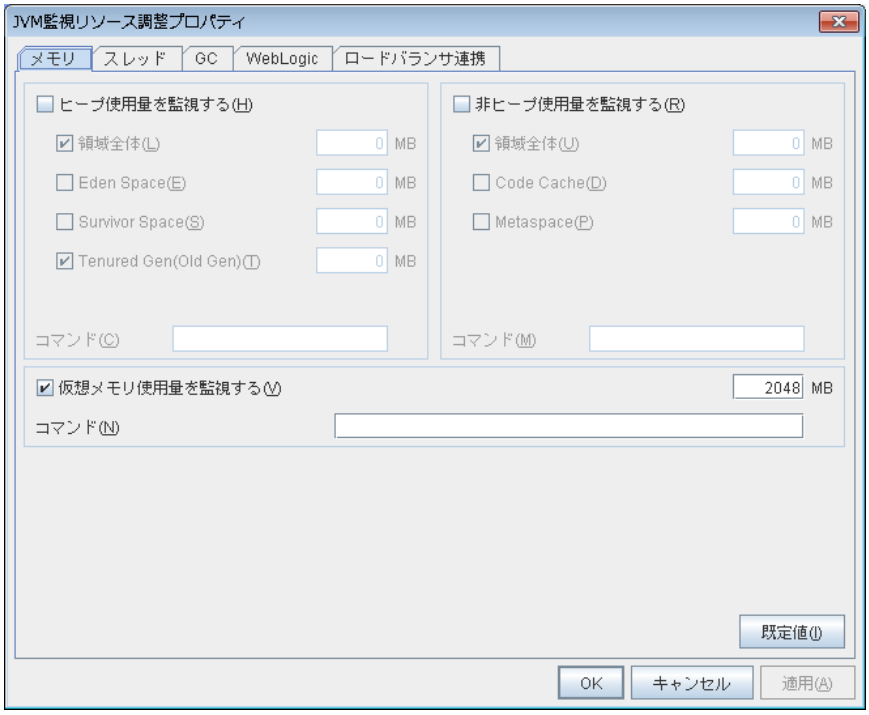
監視対象の Java VM 異常検出時に、実行するコマンドを設定します。異常の原因別に実行するコマンドおよび引数の指定が可能です。絶対パスで指定してください。また、実行ファイル名は二重引用符(“”)で括ってください。例)”/usr/local/bin/command” arg1 arg2
ここでは監視対象 Java VM の Java ヒープ領域、Java 非ヒープ領域、仮想メモリ使用量における異常検出時に、実行するコマンドを設定します。
「異常検出時に障害原因別にコマンドを実行するには」も参照してください。

既定値 : なし

既定値

[既定値]ボタンをクリックすると全ての項目に既定値が設定されます。

メモリタブ（[JVM種別]で[Oracle Java(usage monitoring)] 選択時）



ヒープ使用量を監視する

監視対象の Java VM が使用する Java ヒープ領域の使用量の監視設定をします。

- チェックボックスがオン
  監視します。
- チェックボックスがオフ(既定値)
  監視しません。

領域全体(1～102400)

監視対象の Java VM が使用する Java ヒープ領域の使用量のしきい値を設定します。

既定値 ： 0[MB]

Eden Space(1～102400)

監視対象の Java VM が使用する Java Eden Space の使用量のしきい値を設定します。GC 方式として G1 GC を指定している場合、G1 Eden Space と読み替えてください。

既定値 ： 0[MB]

Survivor Space(1～102400)

監視対象の Java VM が使用する Java Survivor Space の使用量のしきい値を設定します。GC 方式として G1 GC を指定している場合、G1 Survivor Space と読み替えてください。

既定値 ： 0[MB]

Tenured Gen(1～102400)

監視対象の Java VM が使用する Java Tenured(Old) Gen 領域の使用量のしきい値を設定します。GC 方式として G1 GC を指定している場合、G1 Old Gen と読み替えてください。
既定値 ： 0[MB]

非ヒープ使用量を監視する

監視対象の Java VM が使用する Java 非ヒープ領域の使用量の監視設定をします。

- チェックボックスがオン
  監視します。
- チェックボックスがオフ(既定値)
  監視しません。

領域全体(1～102400)

監視対象の Java VM が使用する Java 非ヒープ領域の使用量のしきい値を設定します。
既定値 ： 0[MB]

Code Cache(1～102400)

監視対象の Java VM が使用する Java Code Cache 領域の使用量のしきい値を設定します。
既定値 ： 0[MB]

Metaspace(1～102400)

監視対象の Java VM が使用する Metaspace 領域の使用量のしきい値を設定します。
既定値 ： 0[MB]

仮想メモリ使用量を監視する(1～3072)

監視対象の Java VM が使用する仮想メモリ使用量のしきい値を設定します。JVM モニタリソースでは、監視対象の Java VM が 64bit プロセスの場合は仮想メモリ使用量の監視は対応しておりません。そのため、監視対象の Java VM が 64bit プロセスの場合はチェックボックスをオフにしてください。
既定値 ： 2048[MB]

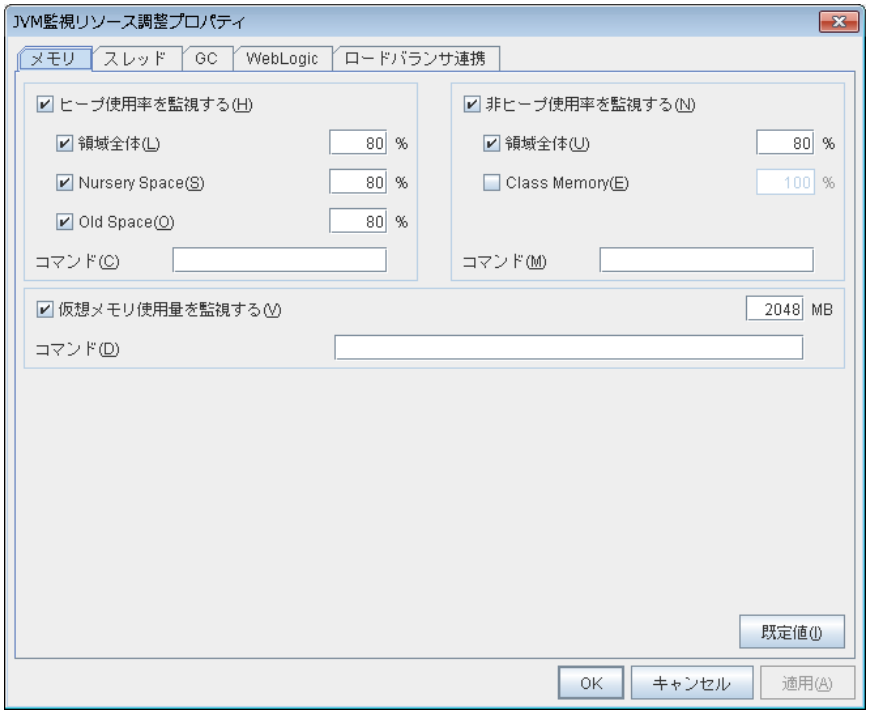
コマンド(255 バイト以内)

監視対象の Java VM 異常検出時に、実行するコマンドを設定します。異常の原因別に実行するコマンドおよび引数の指定が可能です。絶対パスで指定してください。また、実行ファイル名は二重引用符(“”)で括ってください。例)”/usr/local/bin/command” arg1 arg2
ここでは監視対象 Java VM の Java ヒープ領域、Java 非ヒープ領域、仮想メモリ使用量における異常検出時に、実行するコマンドを設定します。
「異常検出時に障害原因別にコマンドを実行するには」も参照してください。
既定値 ： なし

既定値

[既定値]ボタンをクリックすると全ての項目に既定値が設定されます。

メモリタブ（Oracle JRockit選択時）



[JVM 種別]で[JRockit]選択時のみ表示されます。

ヒープ使用率を監視する

監視対象の Java VM が使用する Java ヒープ領域の使用率の監視設定をします。

- チェックボックスがオン(既定値)
  監視します。
- チェックボックスがオフ
  監視しません。

領域全体(1～100)

監視対象の Java VM が使用する Java ヒープ領域の使用率のしきい値を設定します。

既定値 ： 80[%]

Nursery Space(1～100)

監視対象の JRockit JVM が使用する Java Nursery Space の使用率のしきい値を設定します。

既定値 ： 80[%]

Old Space(1～100)

監視対象の JRockit JVM が使用する Java Old Space の使用率のしきい値を設定します。

既定値 ： 80[%]

非ヒープ使用率を監視する

監視対象の Java VM が使用する Java 非ヒープ領域の使用率の監視設定をします。

- チェックボックスがオン(既定値)
  監視します。
- チェックボックスがオフ
  監視しません。

領域全体(1～100)

監視対象の Java VM が使用する Java ヒープ領域の使用率のしきい値を設定します。

既定値 : 80[%]

Class Memory(1～100)

監視対象の JRockit JVM が使用する Java Class Memory の使用率のしきい値を設定します。

既定値 : 100[%]

仮想メモリ使用量を監視する(1～3072)

監視対象の Java VM が使用する仮想メモリ使用量のしきい値を設定します。JVM モニタリソースでは、監視対象の Java VM が 64bit プロセスの場合は仮想メモリ使用量の監視は対応しておりません。そのため、監視対象の Java VM が 64bit プロセスの場合はチェックボックスをオフにしてください。

既定値 : 2048[MB]

コマンド(255 バイト以内)

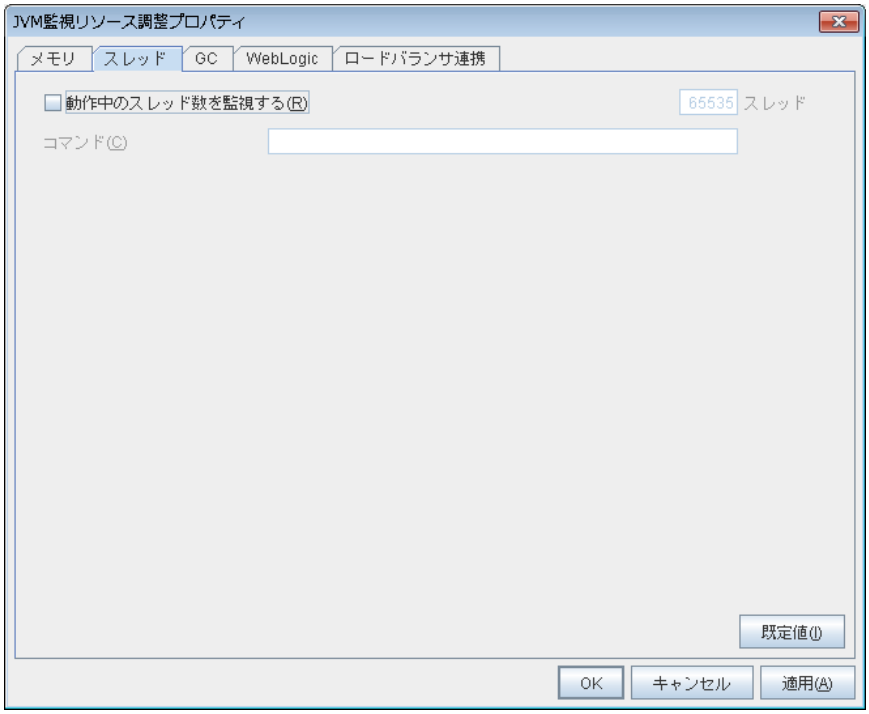
監視対象の Java VM 異常検出時に、実行するコマンドを設定します。異常の原因別に実行するコマンドおよび引数の指定が可能です。絶対パスで指定してください。また、実行ファイル名は二重引用符(“”)で括ってください。例)”/usr/local/bin/command” arg1 arg2
ここでは監視対象 Java VM の Java ヒープ領域、Java 非ヒープ領域、仮想メモリ使用量における異常検出時に、実行するコマンドを設定します。
「異常検出時に障害原因別にコマンドを実行するには」も参照してください。

既定値 : なし

既定値

[既定値]ボタンをクリックすると全ての項目に既定値が設定されます。

スレッドタブ



動作中のスレッド数を監視する(1～65535)

監視対象の Java VM で現在動作中のスレッド上限数のしきい値を設定します。

既定値 ： 65535[スレッド]

コマンド(255 バイト以内)

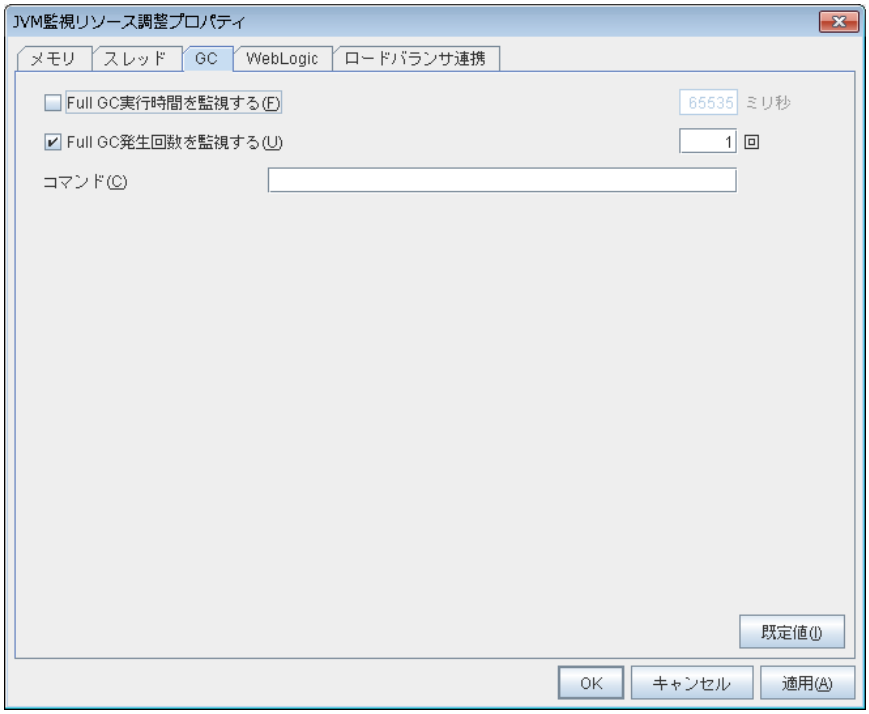
監視対象の Java VM 異常検出時に、実行するコマンドを設定します。異常の原因別に実行するコマンドおよび引数の指定が可能です。絶対パスで指定してください。また、実行ファイル名は二重引用符(“”)で括ってください。例)”/usr/local/bin/command” arg1 arg2
ここでは監視対象 Java VM で現在動作中のスレッド数における異常検出時に、実行するコマンドを設定します。
「異常検出時に障害原因別にコマンドを実行するには」も参照してください。

既定値 ： なし

既定値

[既定値]ボタンをクリックすると全ての項目に既定値が設定されます。

GCタブ



Full GC 実行時間を監視する(1～65535)

監視対象の Java VM において、前回計測以降の Full GC 実行時間のしきい値を設定します。Full GC 実行時間とは、前回計測以降の Full GC 発生回数で割った平均値です。

前回計測以降の Full GC 実行時間 3000 ミリ秒、Full GC 発生回数 3 回の場合を異常と判定したい場合、1000 ミリ秒以下を設定してください。

既定値 ： 65535[ミリ秒]

Full GC 発生回数を監視する(1～65535)

監視対象の Java VM において、前回計測以降の Full GC 発生回数のしきい値を設定します。

既定値 ： 1(回)

コマンド(255 バイト以内)

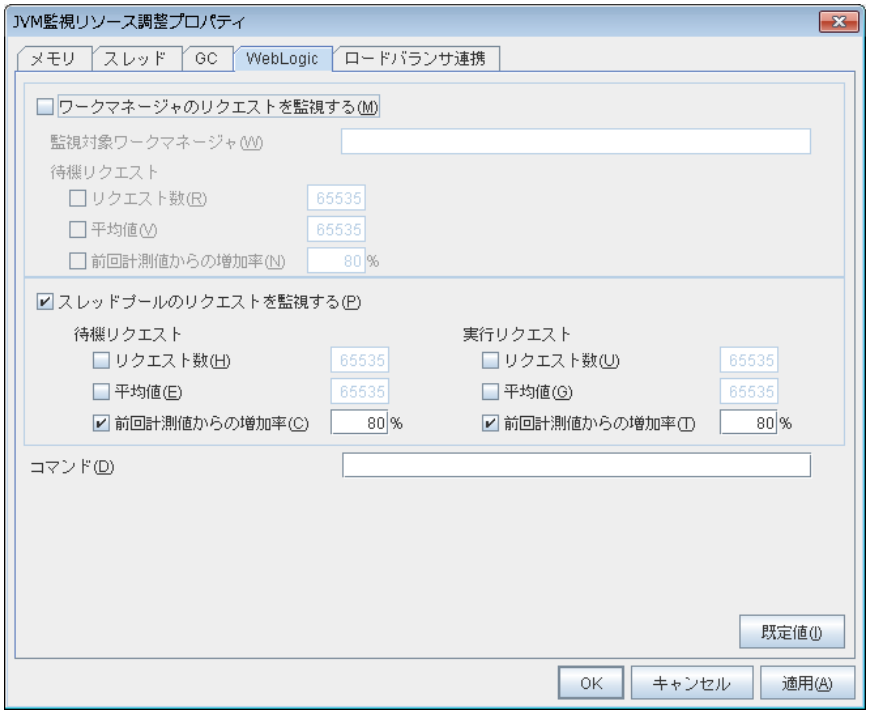
監視対象の Java VM 異常検出時に、実行するコマンドを設定します。異常の原因別に実行するコマンドおよび引数の指定が可能です。絶対パスで指定してください。また、実行ファイル名は二重引用符(“”)で括ってください。例)”/usr/local/bin/command” arg1 arg2
ここでは監視対象 Java VM の Full GC 実行時間や Full GC 発生回数における異常検出時に、実行するコマンドを設定します。
「異常検出時に障害原因別にコマンドを実行するには」も参照してください。

既定値 ： なし

既定値

[既定値]ボタンをクリックすると全ての項目に既定値が設定されます。

WebLogicタブ



[監視対象]で[WebLogic Server]選択時のみ表示されます。

ワークマネージャのリクエストを監視する

WebLogic Server でワークマネージャの待機リクエスト状態の監視設定をします。

- チェックボックスがオン
  監視します。
- チェックボックスがオフ(既定値)
  監視しません。

監視対象ワークマネージャ

監視対象の WebLogic Server に対して監視したいアプリケーションのワークマネージャ名を設定します。ワークマネージャ監視を実施する場合、必ず設定してください。

*App*1[*WM*1,*WM*2,…];*App*2[*WM*1,*WM*2,…];…

App と *WM* にて指定可能な文字は ASCII 文字です。(Shift_JIS コード 0x005C と 0x00A1～0x00DF を除く)

アプリケーション アーカイブのバージョンを持つアプリケーションを指定する場合、*App* には「アプリケーション名#バージョン」を指定してください。

アプリケーション名に"["や"]"が付いている場合、"["や"]"の直前に「¥¥」を追加してください。

(例) アプリケーション名が app[2]の場合、app¥¥[2¥¥]

既定値 ： なし

リクエスト数(1～65535)

監視対象の WebLogic Server のワークマネージャにおいて、待機リクエスト数のしきい値を設定します。

既定値 ： 65535

平均値(1～65535)

監視対象のWebLogic Serverのワークマネージャにおいて、待機リクエスト数の平均値のしきい値を設定します。

既定値 ： 65535

前回計測値からの増加率(1～1024)

監視対象の WebLogic Server のワークマネージャにおいて、待機リクエスト数の前回計測以降の増分に対するしきい値を設定します。

既定値 ： 80[%]

スレッドプールのリクエストを監視する

監視対象の WebLogic Server のスレッドプールにおいて、待機リクエスト数 、実行リクエスト数の監視設定をします。リクエスト数とは、WebLogic Server 内部で処理待ちや実行した HTTP リクエスト数や、EJB の呼び出しや WebLogic Server 内部で行われる処理のリクエスト数を含みます。ただし、増加しても異常な状態とは判断できません。JVM 統計ログの採取をする場合に指定してください。

- チェックボックスがオン(既定値)
  監視します。
- チェックボックスがオフ
  監視しません。

待機リクエスト リクエスト数(1～65535)

待機リクエスト数のしきい値を設定します。

既定値 ： 65535

待機リクエスト 平均値(1～65535)

待機リクエスト数の平均値のしきい値を設定します。

既定値 ： 65535

待機リクエスト 前回計測値からの増加率(1～1024)

待機リクエスト数の前回計測以降の増分に対するしきい値を設定します。

既定値 ： 80[%]

実行リクエスト リクエスト数(1～65535)

単位時間あたりに実行したリクエスト数のしきい値を設定します。

既定値 ： 65535

実行リクエスト 平均値(1～65535)

単位時間あたりに実行したリクエスト数の平均値のしきい値を設定します。

既定値 : 65535

実行リクエスト 前回計測値からの増加率(1～1024)

単位時間あたりに実行したリクエスト数の前回計測以降の増分に対するしきい値を設定します。
既定値 : 80[%]

コマンド(255 バイト以内)

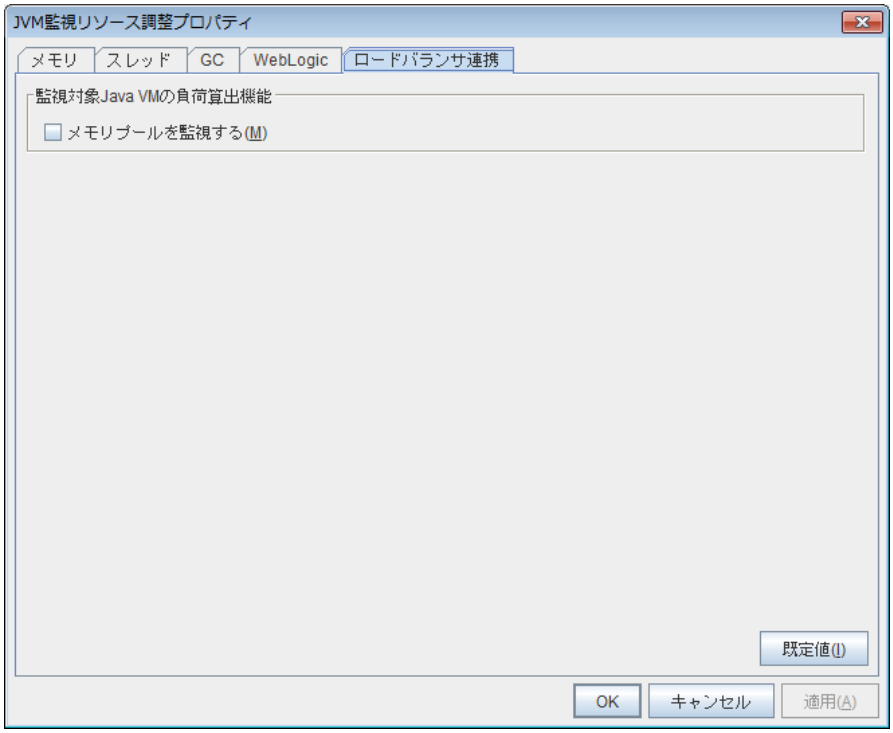
監視対象の Java VM 異常検出時に、実行するコマンドを設定します。異常の原因別に実行するコマンドおよび引数の指定が可能です。絶対パスで指定してください。また、実行ファイル名は二重引用符(“”)で括ってください。例)”/usr/local/bin/command” arg1 arg2
ここでは WebLogic Server のワークマネージャのリクエストやスレッドプールのリクエストにおける異常検出時に、実行するコマンドを設定します。
「異常検出時に障害原因別にコマンドを実行するには」も参照してください。

既定値 : なし

既定値

[既定値]ボタンをクリックすると全ての項目に既定値が設定されます。

ロードバランサ連携タブ



ロードバランサ種別に[BIG-IP LTM]以外を選択した場合、本画面が表示されます。

メモリプールを監視する

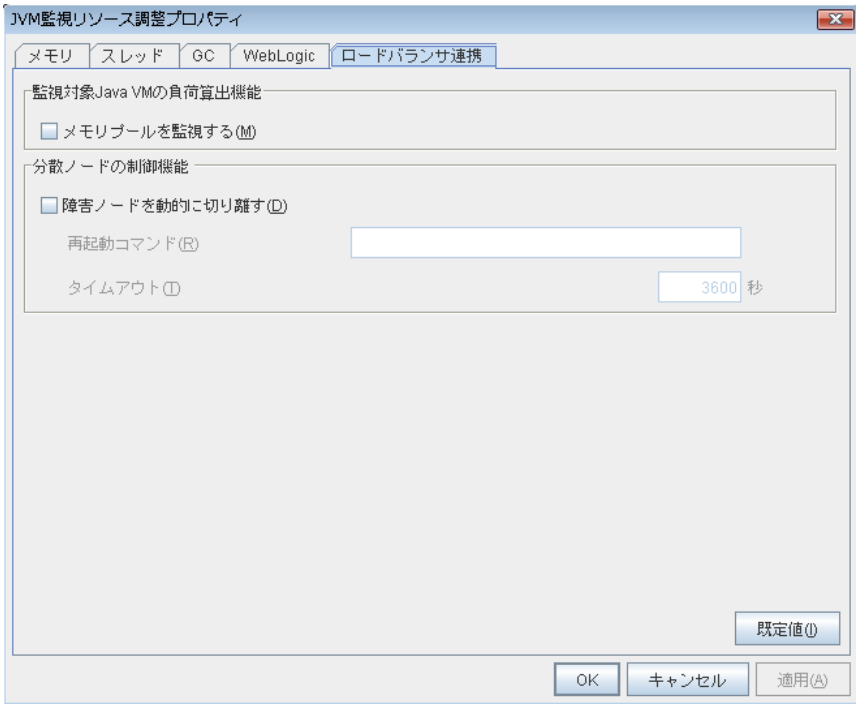
ロードバランサに動的負荷情報を通知する際、メモリプールを監視対象とするかを設定します。

- チェックボックスがオン
  監視します。
- チェックボックスがオフ(既定値)
  監視しません。

既定値

[既定値]ボタンをクリックすると全ての項目に既定値が設定されます。

ロードバランサ連携タブ(BIG-IP LTM の場合)



ロードバランサ種別に[BIG-IP LTM]を選択した場合、本画面が表示されます。

メモリプールを監視する

ロードバランサに動的負荷情報を通知する際、メモリプールを監視対象とするかを設定します。

- チェックボックスがオン
  監視します。
- チェックボックスがオフ(既定値)
  監視しません。

障害ノードを動的に切り離す

監視対象の障害状態を検出すると(例:採取情報が設定しているしきい値を超えている)、BIG-IP LTM の分散ノードのステータスを enable から disable に更新するかを設定します。

- チェックボックスがオン
  enableからdisableに更新します。
- チェックボックスがオフ(既定値)
  更新しません。

再起動コマンド

分散ノードのコネクション数が 0 になるまで待ち合わせた後、実行したいコマンドを絶対パスで指定します。常駐監視の場合かつ監視対象の障害検出時に、監視対象を再起動したい場合に有効です。再起動コマンドは JVM モニタリソース間で共通の値を設定してください。

タイムアウト(0～2592000)

分散ノードのステータスを enable から disable にした後、分散ノードのコネクション数が 0 になるまで待ち合わせるタイムアウト時間を設定します。タイムアウトした場合は、[再起動コマンド]は実行しません。

既定値 ： 3600[秒]

既定値

[既定値]ボタンをクリックすると[メモリプールを監視する]、[障害ノードを動的に切り離す]、[タイムアウト]の項目に既定値が設定されます。

JVM モニタリソースの注意事項

JVM モニタリソースを作成する前にクラスタプロパティの[JVM 監視]タブの[Java インストールパス]を前もって設定しておく必要があります。

監視対象のリソースには、WebLogic Server や WebOTX など Java VM 上で動作するアプリケーションサーバを指定してください。JVM モニタリソースの活性後、Java Resource Agent は監視を開始しますが、JVM モニタリソースの活性直後に監視対象(WebLogic Server や WebOTX)がすぐに動作できない場合は、[監視開始待ち時間]で調整してください。

[監視(共通)]-[リトライ回数]の設定は無効です。異常の検出を遅らせたい場合は、[クラスタ]プロパティ-[JVM 監視]タブ-[リソース計測設定]-[共通]-[リトライ回数]の設定を変更してください。

JVM モニタリソースの監視方法

JVM モニタリソースは、以下の監視を行います。

JMX(Java Management Extensions)を利用して、アプリケーションサーバの監視を実行します。

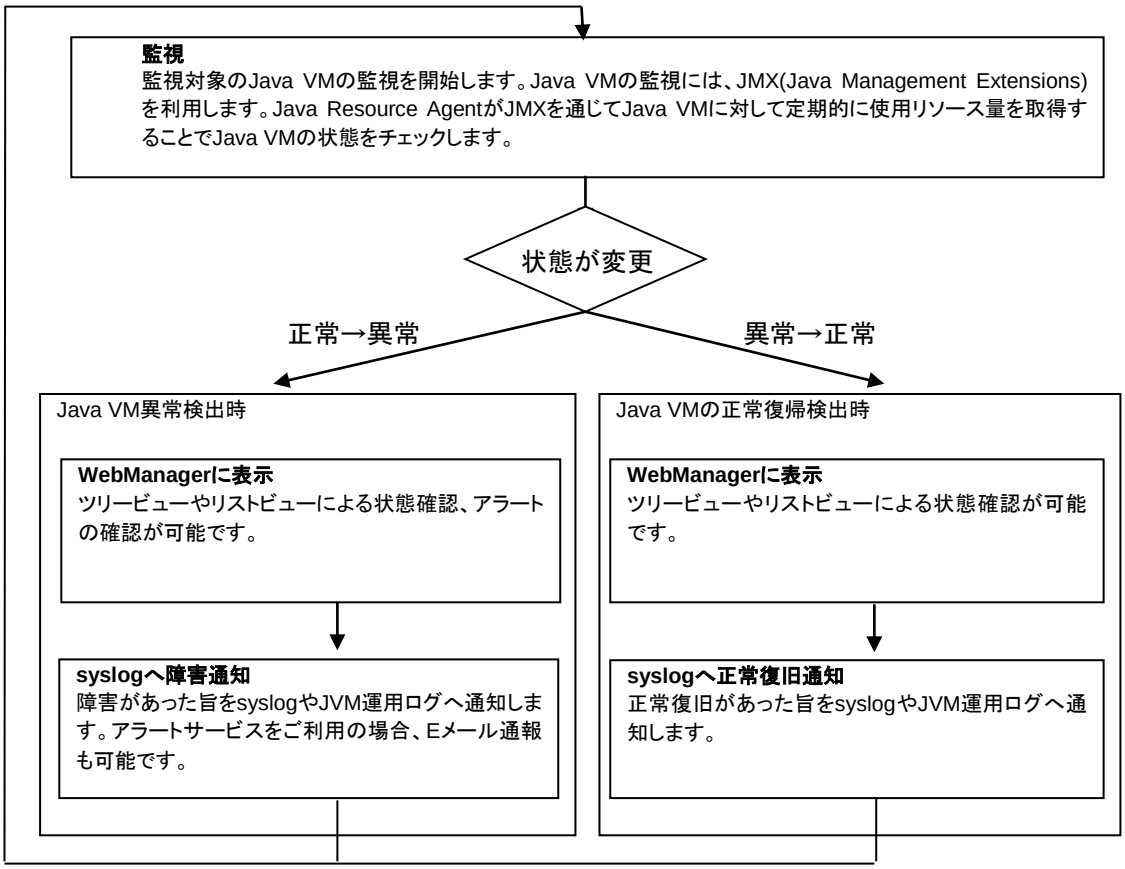
監視の結果、以下の場合に異常とみなします。

- 監視対象の Java VM やアプリケーションサーバに接続できない場合
- 取得した Java VM やアプリケーションサーバのリソース使用量がお客様定義のしきい値を規定回数(異常判定しきい値)連続して超えた場合

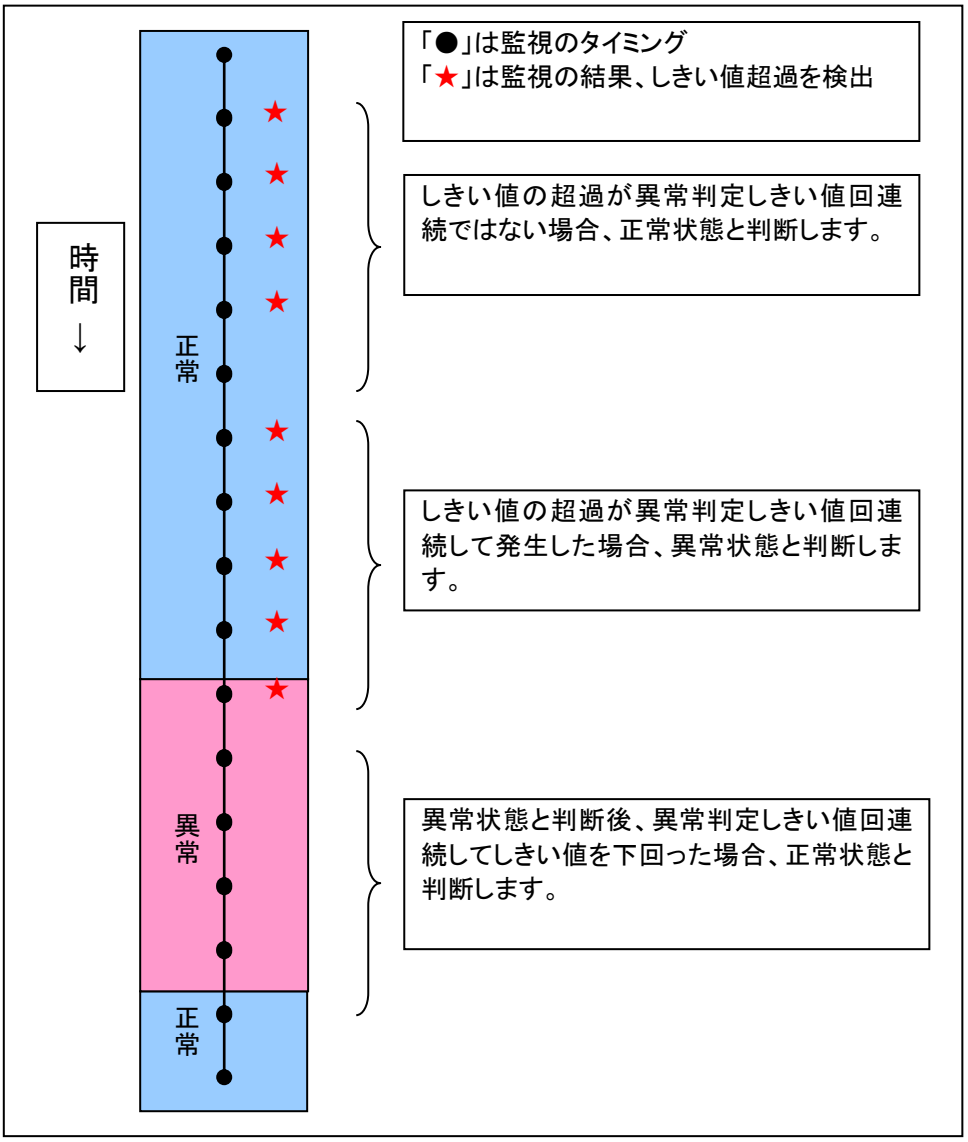
監視の結果、以下の場合に異常から正常へ復帰したとみなします。

- 取得した Java VM やアプリケーションサーバのリソース使用量がお客様定義のしきい値を異常判定しきい値回連続して下回った場合

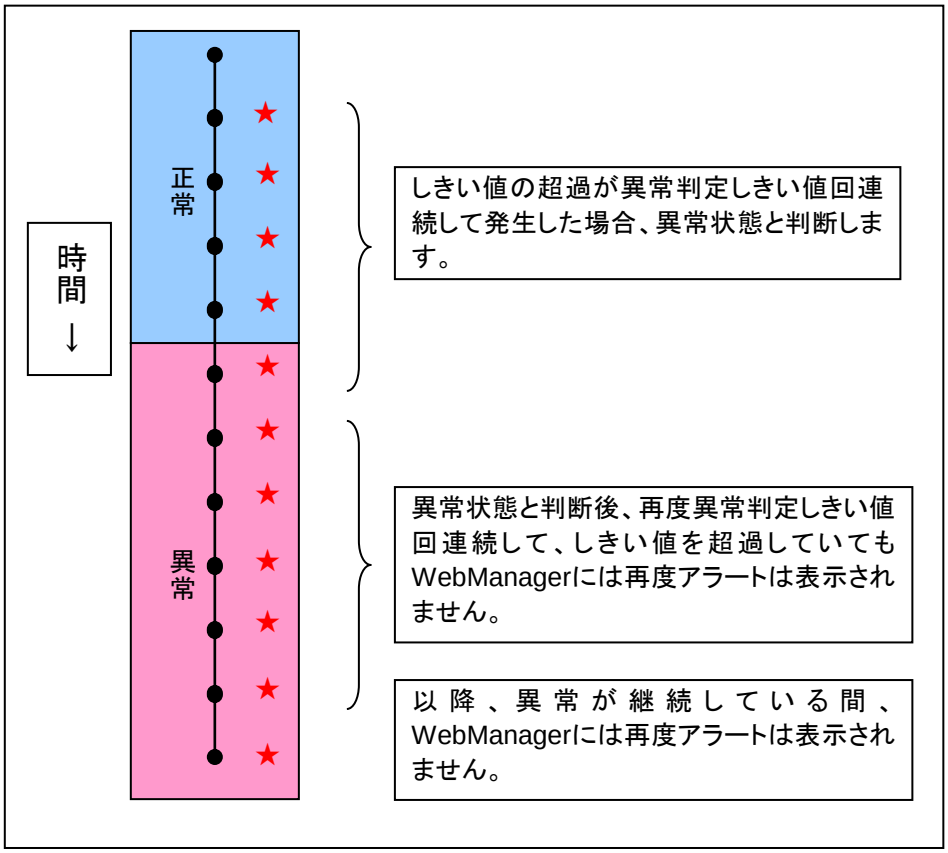
注: WebManager [ツール]メニューの[クラスタログ収集]では、監視対象(WebLogic Server や WebOTX)の設定ファイルおよびログファイルは収集されません。



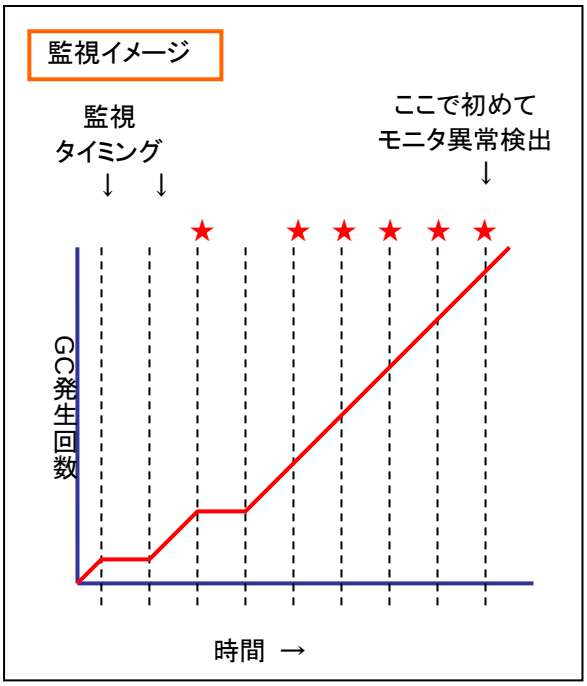
基本的なしきい値超過時の動作は以下の通りです。



異常が継続する場合は以下の通りです



Full GC(Garbage Collection)を監視する場合を例に説明します。
JVMモニタリソースは、異常判定しきい値回連続してFull GCが発生すると、モニタ異常を検出します。以下の「★」はJVM監視がFull GCを検出した状態を示しており、異常判定しきい値を5回に設定した例です。
Full GCはシステムに与える影響が大きいため、異常判定しきい値は1回に設定することを推奨します。



ロードバランサと連携するには(ヘルスチェック機能)

対象ロードバランサ:HTML ファイルへのヘルスチェック機能をもつロードバランサ

JVM モニタリソースは、ロードバランサ連携を行うことが可能です。以降、監視対象のアプリケーションサーバが WebOTX として説明します。ロードバランサ連携は、ヘルスチェック機能、および監視対象 Java VM の負荷算出機能を提供します。BIG-IP Local Traffic Manager と連携する場合は、「BIG-IP Local Traffic Manager と連携するには」を参照してください。

分散ノードとは負荷分散対象サーバ、分散ノードモジュールとは各分散ノードにインストールするモジュールです。分散ノードモジュールは Express5800/LB400*、MIRACLE LoadBalancer に含まれます。Express5800/LB400*の場合は『Express5800/LB400* ユーザーズガイド(ソフトウェア編)』、Express5800/LB400*以外のロードバランサは各マニュアルを参照してください。

本機能を使用するには、Builder のクラスタプロパティ→JVM 監視タブ→ロードバランサ連携設定ダイアログで設定を行ってください。ロードバランサのヘルスチェック機能と連携します。

該当サーバがロードバランサの負荷分散システムを構築している場合、JVM 監視は WebOTX の障害状態を検出すると(例:採取情報が設定しているしきい値を超えている)、[HTML ファイル名]で設定している HTML ファイルを[HTML リネーム先ファイル名]で設定しているファイル名にリネームします。

JVM 監視は HTML ファイル名をリネーム後、待機時間である 20 秒間待ち合わせます。待機するのはロードバランサが分散ノードを切り離す前に WebOTX を再起動してしまうことを防止するためです。

JVM 監視は WebOTX 再起動後に WebOTX の正常状態を検出すると(例:接続後の採取情報が設定しているしきい値を超えていない)、[HTML リネーム先ファイル名]で設定しているHTML ファイルを[HTML ファイル名]で設定しているファイル名に戻します。

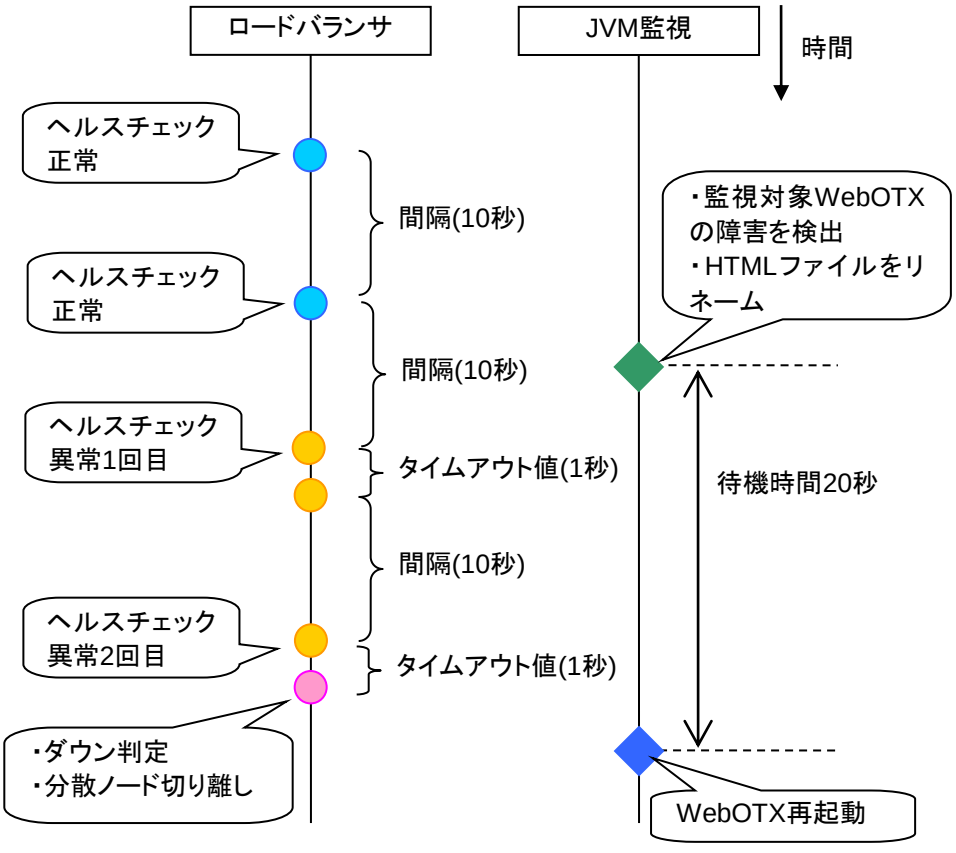
ロードバランサは定期的に HTML ファイルに対しヘルスチェックを実行していますが、ヘルスチェックが失敗すると分散ノードをダウンと判定し、ロードバランサは分散ノードの切り離しを実行します。Express5800/LB400*の場合、ヘルスチェックの間隔、ヘルスチェックのタイムアウト、ヘルスチェックにてノードダウンと判定するまでのリトライ回数は、それぞれロードバランサの[ManagementConsole]から[LoadBalancer]→[システム情報]内のヘルスチェック(分散ノード)間隔パラメータ、HTTP ヘルスチェックタイムアウト値パラメータ、ヘルスチェック(分散ノード)回数パラメータで設定を行ってください。Express5800/LB400*以外のロードバランサの設定は、各マニュアルを参照してください。

各パラメータは以下を参考に設定してください。

待機時間 20 秒 ＞＝ (ヘルスチェック(分散ノード)間隔 ＋ HTTP ヘルスチェックタイムアウト値) × ヘルスチェック(分散ノード)回数

ロードバランサのJVM監視ヘルスチェック機能設定

ヘルスチェック(分散ノード)間隔	10秒
HTTPヘルスチェックタイムアウト値	1秒
ヘルスチェック(分散ノード)回数	2回



ロードバランサと連携するには(監視対象Java VMの負荷算出機能)

対象ロードバランサ:Express5800/LB400*, MIRACLE LoadBalancer

JVM モニタリソースは、ロードバランサ連携を行うことが可能です。以降、監視対象のアプリケーションサーバが WebOTX として説明します。ロードバランサ連携は、ヘルスチェック機能、および監視対象 Java VM の負荷算出機能を提供します。

BIG-IP Local Traffic Manager と連携する場合は、「BIG-IP Local Traffic Manager と連携するには」を参照してください。

分散ノードとは負荷分散対象サーバ、分散ノードモジュールとは各分散ノードにインストールするモジュールです。分散ノードモジュールは Express5800/LB400*、MIRACLE LoadBalancer に含まれます。Express5800/LB400*の場合は『Express5800/LB400* ユーザーズガイド(ソフトウェア編)』、Express5800/LB400*以外のロードバランサは各マニュアルを参照してください。

本機能を使用するには、以下の設定が必要です。ロードバランサの CPU 負荷による重み付け機能と連携します。

- [プロパティ]-[監視(固有)]タブ→[調整]プロパティ-[メモリ]ダイアログ-[ヒープ使用率を監視する]-[領域全体]

[プロパティ]-[監視(固有)]タブ→[調整]プロパティ-[ロードバランサ連携]ダイアログ-[メモリプールを監視する

また、以下の手順に従って、各サーバに分散ノードモジュールをインストールした後、ロードバランサ連携用セットアップコマンド clpjra_lbsetup.sh を実行し、分散ノードモジュールに対して設定を行ってください。分散ノードモジュールの設定は、/etc/ha4.d/lbadmin.conf に書き込まれます。

注: root 権限を持つアカウントで実行してください。

1. [CLUSTERPRO インストールフォルダ]/ha/jra/bin/clpjra_lbsetup.sh を実行します。引数は以下の通りです。
(実行例)`clpjra_lbsetup.sh -e 1 -i 120 -t 180`

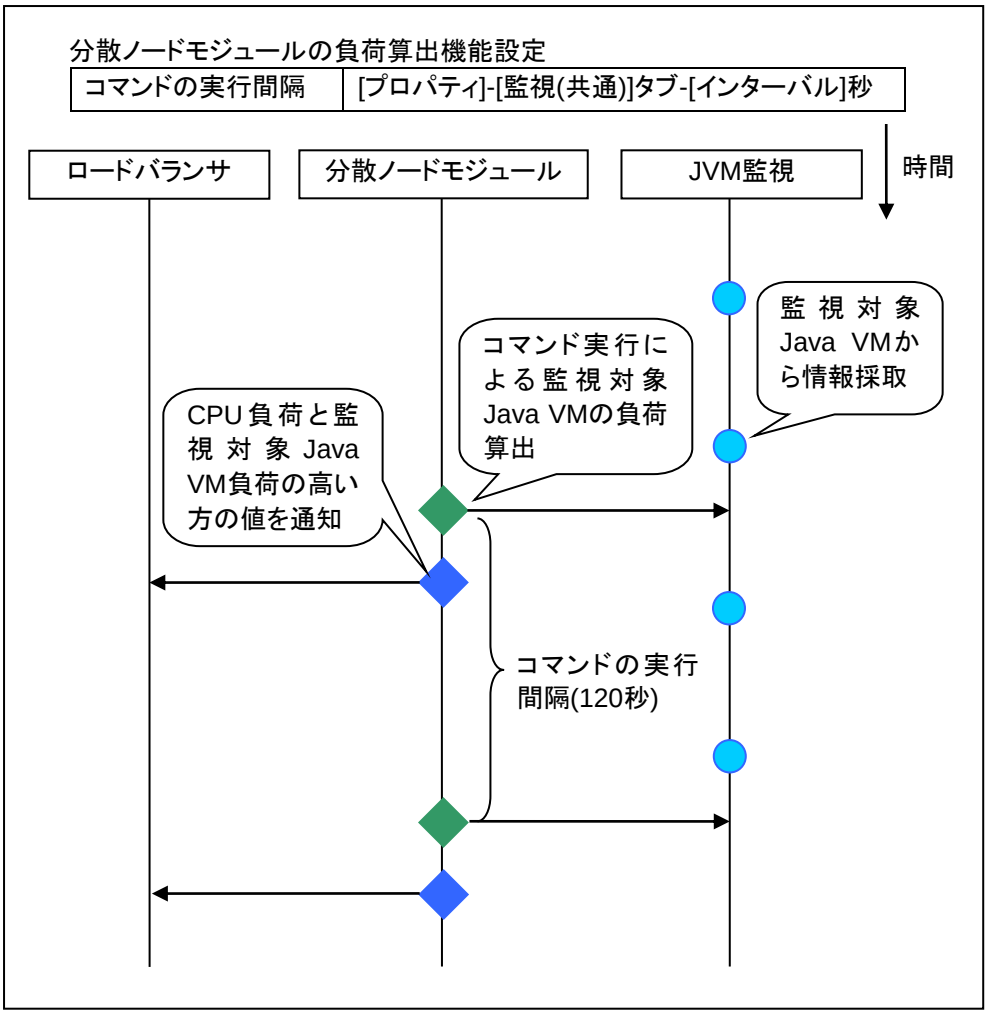
引数	意味	範囲
-e	本機能の無効/有効を指定します。 lbadmin.conf 中のパラメータ名は Enabled です。	0～1 0:機能無効 1:機能有効
-i	監視対象Java VMの負荷を算出するコマンドの実行間隔を秒で指定します。lbadmin.conf 中のパラメータ名は JVMSaverCheckInterval です。	1～2147483646
-t	監視対象Java VMの負荷を算出するコマンドのタイムアウト値を秒で指定します。lbadmin.conf 中のパラメータ名は ActionTimeout です。	1～2147483646

JVM 監視では、採取した Java メモリの情報から監視対象 Java VM の負荷を算出します。

Java VM の負荷は以下の式で計算します。しきい値とは、Java ヒープ領域の全体量に対し、[監視(固有)]タブ-調整プロパティ- [メモリ]タブ-[ヒープ使用率を監視する]-[領域全体]で設定した使用率を掛けた値です。

Java VM の負荷(%)＝現在のメモリ使用量(MB)×100÷しきい値(MB)

JVM 監視が動作するサーバ上にインストールされた分散ノードモジュールでは、コマンドを定期的に実行し、採取した監視対象 Java VM の負荷と、別途採取した CPU 負荷を比較し、高い方の値が CPU 負荷としてロードバランサに通知されます。ロードバランサは分散ノードの CPU 負荷状況に応じて、トラフィック(要求)を最適サーバへ分散させます。



BIG-IP Local Traffic Managerと連携するには

対象ロードバランサ：BIG-IP Local Traffic Manager

JVM モニタリソースは、BIG-IP LTM と連携を行うことが可能です。以降、監視対象のアプリケーションサーバが Tomcat として説明します。BIG-IP LTM との連携は、分散ノードの制御機能、および監視対象 Java VM の負荷算出機能を提供します。

BIG-IP LTM と JVM モニタリソースの連携は、BIG-IP シリーズ API(iControl)により実現しています。

分散ノードとは負荷分散対象サーバ、連携モジュールとは各分散ノードにインストールするモジュールです。連携モジュールは Java Resource Agent に含まれます。

分散ノードの制御機能を使用するには、Builder のクラスタプロパティ→JVM 監視タブ→ロードバランサ連携設定ダイアログ、JVM モニタリソースの[プロパティ]-[監視(固有)]タブ-[調整]プロパティ-[ロードバランサ連携]タブで設定を行ってください。

監視対象 Java VM の負荷算出機能を使用するには、Builder のクラスタプロパティ→JVM 監視タブ→ロードバランサ連携設定ダイアログで設定を行ってください。

BIG-IP LTM 連携におけるエラーメッセージは、JVM 運用ログにも以下の内容が出力されます。詳細は「JVM モニタリソースのログ出力メッセージ」を参照してください。

```
Error: Failed to operate clpjra_bigip.[エラーコード]
```

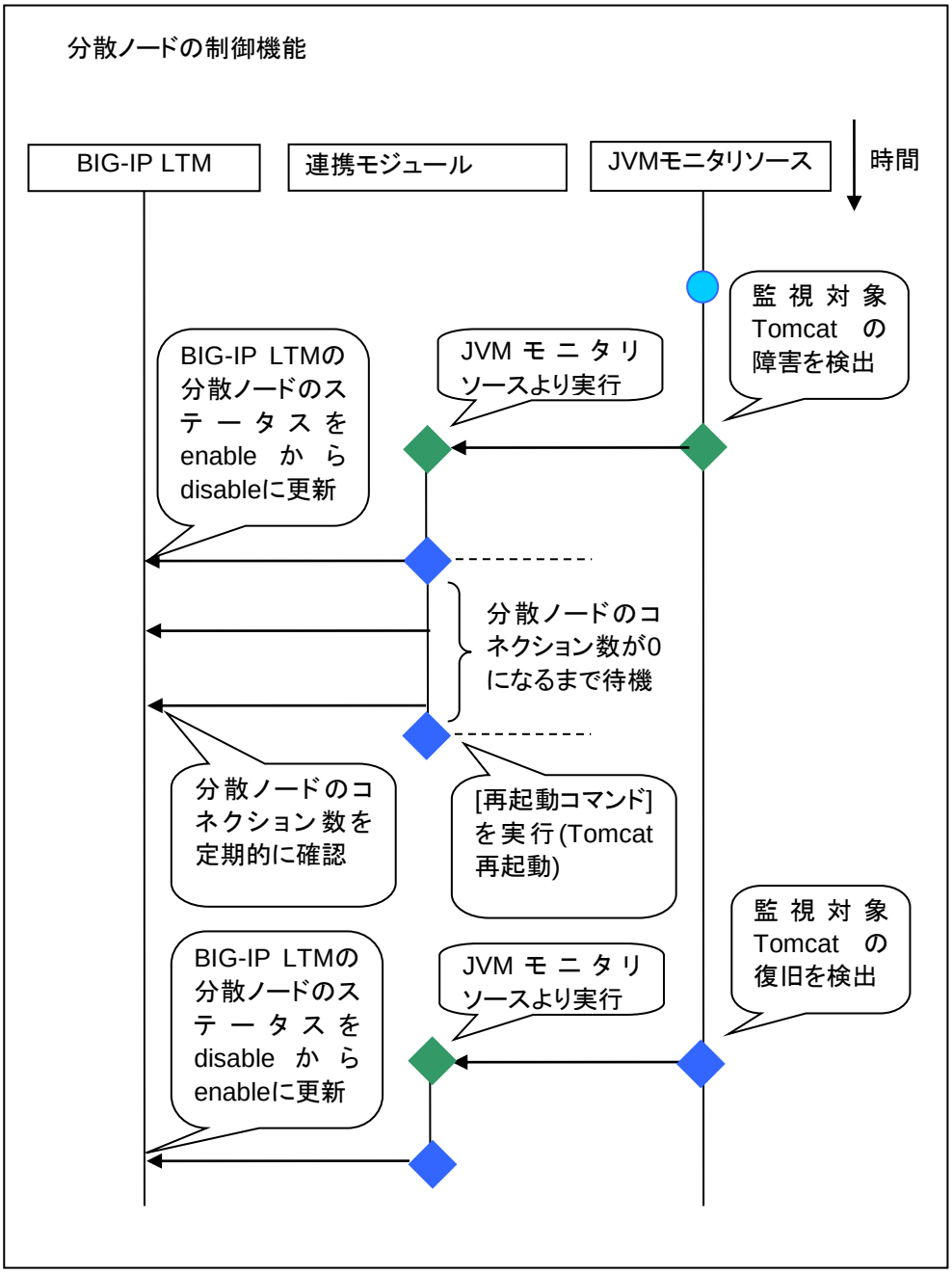
該当サーバが BIG-IP LTM の負荷分散システムを構築している場合、JVM 監視は Tomcat の障害状態を検出すると(例：採取情報が設定しているしきい値を超えている)、iControl を使用し、BIG-IP LTM の分散ノードのステータスを enable から disable に更新します。

JVM 監視は BIG-IP LTM の分散ノードのステータスを更新後、分散ノードのコネクション数が 0 になるまで待ち合わせます。待ち合わせ後、JVM モニタリソースの[プロパティ]-[監視(固有)]タブ-[調整]プロパティ-[ロードバランサ連携]タブで指定した[再起動コマンド]を実行します。JVM モニタリソースの[プロパティ]-[監視(固有)]タブ-[調整]プロパティ-[ロードバランサ連携]タブで指定した[タイムアウト]を経過しても分散ノードのコネクション数が 0 にならない場合、[再起動コマンド]で指定したアクションは実行しません。

JVM 監視は Tomcat の障害復旧を検出すると、iControl を使用し、BIG-IP LTM の分散ノードのステータスを disable から enable に更新します。その際は、JVM モニタリソースの[プロパティ]-[監視(固有)]タブ-[調整]プロパティ-[ロードバランサ連携]タブで指定した[再起動コマンド]で指定したアクションは実行しません。

BIG-IP LTM は分散ノードのステータスが disable の場合、分散ノードをダウンと判定し、BIG-IP LTM は分散ノードの切り離しを実行します。分散ノードの制御機能を使用する場合、BIG-IP LTM 側では、設定は不要です。

BIG-IP LTM の分散ノードのステータス更新は、JVM 監視による障害検出や障害復旧検出を契機としています。そのため、JVM 監視以外を契機としたフェイルオーバの場合、フェイルオーバ後も BIG-IP LTM の分散ノードのステータスは enable である可能性があります。


分散ノードの制御機能
BIG-IP LTM
連携モジュール
JVMモニタリソース
時間
監視対象Tomcatの障害を検出
JVMモニタリソースより実行
BIG-IP LTMの分散ノードのステータスをenableからdisableに更新
分散ノードのコネクション数が0になるまで待機
分散ノードのコネクション数を定期的に確認
[再起動コマンド]を実行(Tomcat再起動)
監視対象Tomcatの復旧を検出
JVMモニタリソースより実行
BIG-IP LTMの分散ノードのステータスをdisableからenableに更新

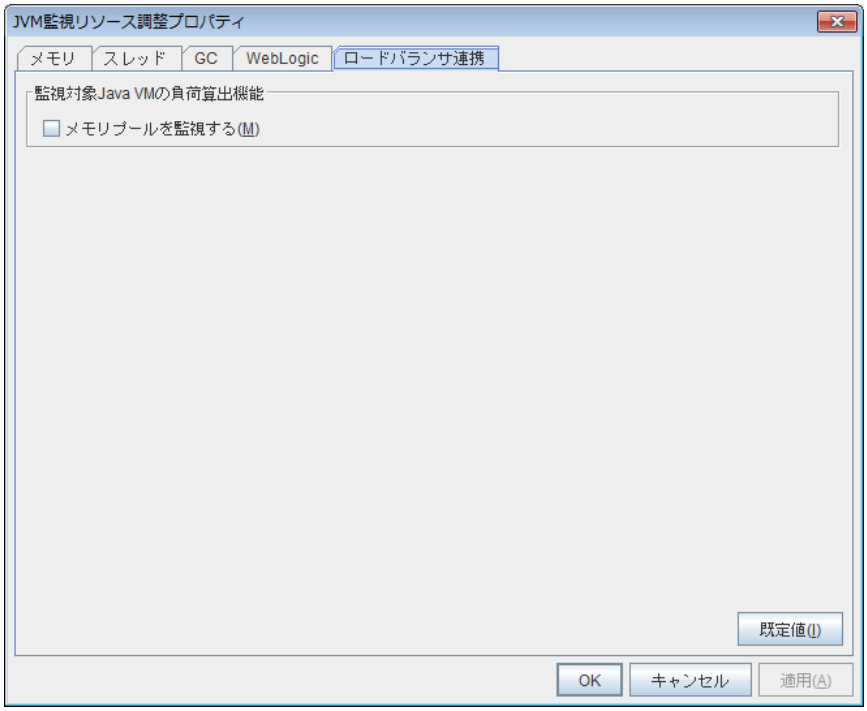
JVM 監視では、採取した Java メモリの情報から監視対象 Java VM の負荷を算出します。Java VM の負荷は以下の式で計算します。Java VM の負荷(%)とは、Javaヒープ領域の全体量に対し、[監視(固有)]タブ-調整プロパティ- [メモリ]タブ-[ヒープ使用率を監視する]-[領域全体]で設定した使用率を掛けた値です。

Java VM の負荷(%)＝現在のメモリ使用量(MB)×100÷しきい値(MB)

JVM 監視が動作するサーバ上にインストールされた連携モジュールでは、コマンドを定期的に実行し、採取した監視対象 Java VM の負荷を BIG-IP LTM に通知します。BIG-IP LTM は分散ノードの Java VM の負荷状況に応じて、トラフィック(要求)を最適なサーバへ分散させます。

CLUSTERPRO 側の設定は、Builder で以下の設定を行ってください。

- JVM モニタリソース



[プロパティ]-[監視(固有)]タブ-[調整]プロパティ-[ロードバランサ連携]タブ
[メモリプールを監視する]をオンにします。

- カスタムモニタリソース

  [プロパティ]-[監視(共通)]タブ
  [監視タイミング]-[常時]のラジオボタンをオンにします。

  [プロパティ]-[監視(固有)]タブ

  [この製品で作成したスクリプト]を選択します。

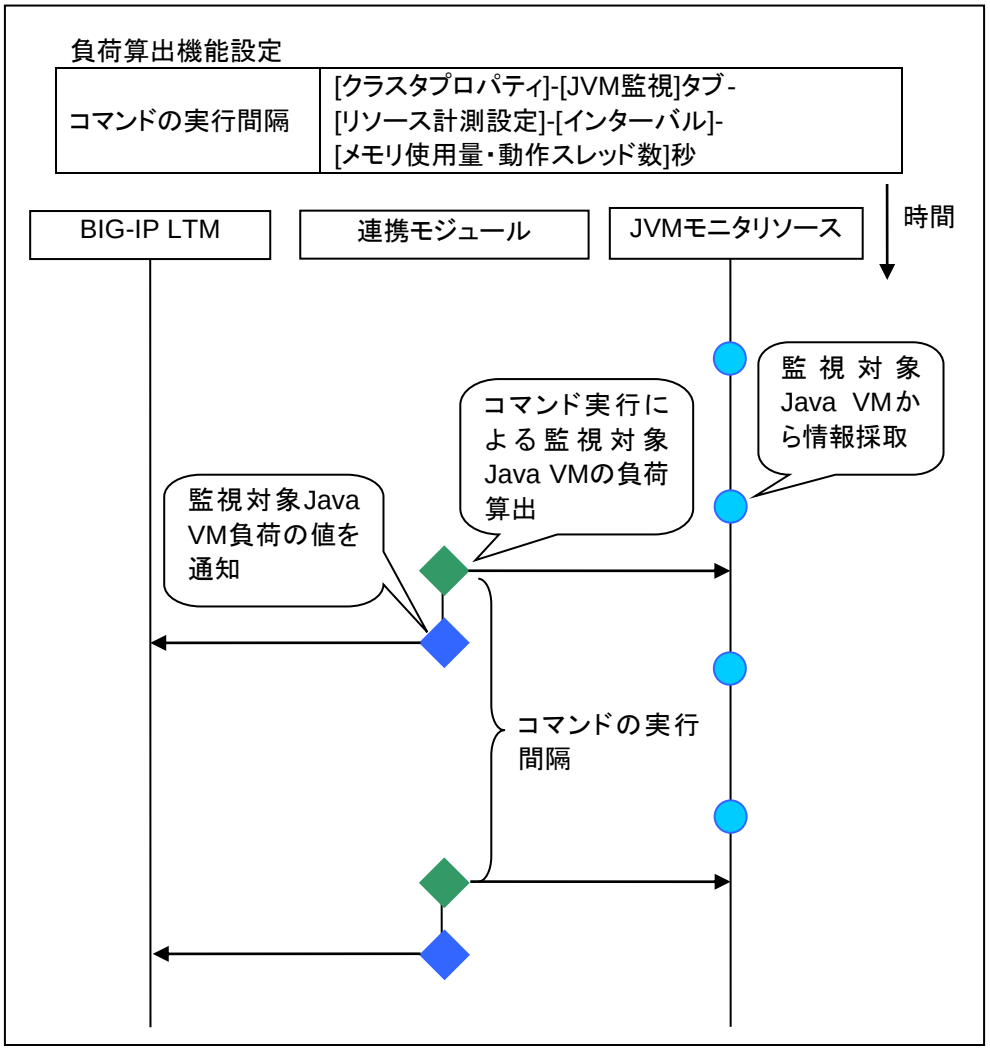
  [ファイル]-[編集]を選択し、下記の太字部分を追記します。

  ```
  ----------------------------------------------------------
  #! /bin/sh
  #************************************************
  #*                  genw.sh                     *
  #************************************************

  ulimit -s unlimited
  ${CLP_PATH}/ha/jra/bin/clpjra_bigip weight
  exit 0
  ----------------------------------------------------------
  ```

  [監視タイプ]-[同期]のラジオボタンをオンにします。

BIG-IP LTM 側の設定は、BIG-IP Configration Utility の LocalTrafic] - [Pools:PoolList] - [該当の pool] - [Members] - [LoadBalancing] - [Load Balancing Method]に[Ratio(node)]を指定してください。


負荷算出機能設定
コマンドの実行間隔
[クラスタプロパティ]-[JVM監視]タブ-
[リソース計測設定]-[インターバル]-
[メモリ使用量・動作スレッド数]秒
BIG-IP LTM
連携モジュール
JVMモニタリソース
時間
監視対象Java VMから情報採取
コマンド実行による監視対象Java VMの負荷算出
監視対象Java VM負荷の値を通知
コマンドの実行間隔

JVM統計ログとは

JVMモニタリソースが収集する監視対象Java VMの統計情報を保存したファイルが、JVM統計ログです。ファイル形式はcsv形式です。

下記の「監視項目」とは、JVMモニタリソースの[プロパティ]-[監視(固有)] タブ内の設定項目を表します。
それぞれの監視項目について、[監視する]をチェックし、かつ閾値を設定した場合に統計情報を採取し、JVM統計ログに情報を出力します。[監視する]をチェックしない場合、および[監視する]をチェックしたが閾値を設定しない場合は、JVM統計ログには情報は出力されません。
監視項目と該当するJVM統計ログは以下の通りです。

監視項目	該当するJVM統計ログ
[メモリ]タブ-[ヒープ使用率を監視する] [メモリ]タブ-[非ヒープ使用率を監視する]	jramemory.stat
[スレッド]タブ-[動作中のスレッド数を監視する]	jrathread.stat
[GC]タブ-[Full GC 実行時間を監視する] [GC]タブ-[Full GC 発生回数を監視する]	jragc.stat
[メモリ]タブ-[仮想メモリ使用量を監視する]	jraruntime.stat
[WebLogic]タブ-[ワークマネージャのリクエストを監視する]	wlworkmanager.stat
[WebLogic]タブ-[スレッドプールのリクエストを監視する]	wlthreadpool.stat

監視対象Java VMのJavaメモリ領域の使用量を確認する(jramemory.stat)

監視対象Java VMのJavaメモリ領域の使用量を記録するログファイルです。ファイル名はログ出力設定のローテーション方式により以下のいずれかになります。

- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[ファイルサイズ]を選択した場合：jramemory<*0から始まる整数*>.stat
- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[時間]を選択した場合：jramemory<*YYYYMMDDhhmm*>.stat

フォーマットは以下の通りです。

No	フォーマット	説明
1	yyyy/mm/dd hh:mm:ss.SSS	ログを記録した日時を示します。
2	半角英数字記号	監視対象Java VMの名称を示します。JVMモニタリソースの[プロパティ]-[監視固有]タブ-[識別名]で設定した値です。
3	半角英数字記号	Javaメモリプールの名称です。詳細は「Javaメモリプール名について」を参照してください。
4	半角英数字記号	Javaメモリプールのタイプです。 Heap、Non-Heap
5	半角数字	Java VMが起動時にOSに要求するメモリ量です。単位はバイトです。(init) 監視対象Java VMの起動時、以下のJava VM起動時オプションでサイズの指定が可能です。 ・HEAP：-Xms ・NON_HEAP パーマネント領域(Perm Gen)：-XX:PermSize ・NON_HEAP コードキャッシュ領域(Code Cache)：-XX:InitialCodeCacheSize

6	半角数字	Java VMが現在使用しているメモリ量です。単位はバイトです。(used)
7	半角数字	Java VMが現在使用することを保証しているメモリ量です。単位はバイトです。(committed) メモリの使用状況により増減しますが、必ずused以上、max以下になります。
8	半角数字	Java VMが使用できる最大メモリ量です。単位はバイトです。(max) 以下のJava VM起動時オプションでサイズの指定が可能です。 ・HEAP:-Xmx ・NON_HEAP パーマネント領域(Perm Gen):-XX:MaxPermSize ・NON_HEAP コードキャッシュ領域(Code Cache):-XX:ReservedCodeCacheSize 例) java -XX:MaxPermSize=128m -XX:ReservedCodeCacheSize=128m javaAP 上記例では NON_HEAP の max は 128m+128m=256m になります。 (注意) -Xms と-Xmxに同じ値を指定すると、(init)>(max)となることがあります。これはHEAPのmaxが、-Xmxの指定によって確保される領域サイズからSurvivor Spaceのサイズの半分を除いたサイズを示すためです。
9	半角数字	計測対象のJava VMが起動してから使用したメモリ量のピーク値です。Javaメモリプールの名称がHEAP、NON_HEAPの場合、Java VMが現在使用しているメモリ量(used)と同じになります。単位はバイトです。
10	半角数字	Javaメモリプールのタイプ(No.4のフィールド)がHEAPの場合、max(No.8のフィールド)×しきい値(%)のメモリ量です。単位はバイトです。 Javaメモリプールのタイプが HEAP 以外の場合、0固定です。

監視対象Java VMのスレッド稼働状況を確認する(jrathread.stat)

監視対象Java VMのスレッド稼働状況を記録するログファイルです。ファイル名はログ出力設定のローテーション方式により以下のいずれかになります。

- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[ファイルサイズ]を選択した場合:jrathread<*0から始まる整数*>.stat
- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[時間]を選択した場合:jrathread<*YYYYMMDDhhmm*>.stat

フォーマットは以下の通りです。

No	フォーマット	説明
1	yyyy/mm/dd hh:mm:ss.SSS	ログを記録した日時を示します。

2	半角英数字記号	監視対象Java VMの名称を示します。JVMモニタリソースの[プロパティ]-[監視固有]タブ-[識別名]で設定した値です。
3	半角英数字記号	監視対象Java VMで現在実行中のスレッド数を示します。
4	[半角数字: 半角数字:...]	監視対象Java VMでデッドロックしているスレッドIDを示します。デッドロック数分IDを繰り返します。
5	半角英数字記号	監視対象Java VMでデッドロックしているスレッドの詳細情報を示します。スレッド数分、以下の形式で繰り返します。 スレッド名，スレッドID，スレッド状態，UserTime, CpuTime, WaitedCount, WaitedTime, isInNative, isSuspended <改行> stacktrace<改行> : stacktrace<改行> stacktrace=ClassName, FileName, LineNumber, MethodName, isNativeMethod

監視対象Java VMのGC稼働状況を確認する(jragc.stat)

監視対象Java VMのGC稼働状況を記録するログファイルです。ファイル名はログ出力設定のローテーション方式により以下のいずれかになります。

- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[ファイルサイズ]を選択した場合：jragc<*0から始まる整数*>.stat
- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[時間]を選択した場合：jragc<*YYYYMMDDhhmm*>.stat

JVMモニタリソースではコピーGCとFull GCの2種類のGCの情報を出力しています。
JVMモニタリソースでは、Oracle Javaの場合は以下のGCについて、Full GCとして発生回数の増分をカウントしています。

- MarksweepCompact
- MarkSweepCompact
- PS Marksweep
- ConcurrentMarkSweep

フォーマットは以下の通りです。

No	フォーマット	説明
1	yyyy/mm/dd hh:mm:ss.SSS	ログを記録した日時を示します。
2	半角英数字記号	監視対象Java VMの名称を示します。JVMモニタリソースの[プロパティ]-[監視固有]タブ-[識別名]で設定した値です。
3	半角英数字記号	監視対象Java VM のGC名称を示します。 **監視対象Java VMがOracle Javaの場合** 以下があります。 Copy MarksweepCompact MarkSweepCompact PS Scavenge PS Marksweep ParNew

		ConcurrentMarkSweep **監視対象Java VMがOracle JRockitの場合** 以下があります。 Garbage collection optimized for throughput Old Collector Garbage collection optimized for short pausetimes Old Collector Garbage collection optimized for deterministic pausetimes Old Collector Static Collector Static Old Collector Garbage collection optimized for throughput Young Collector
4	半角数字	監視対象Java VMの起動直後から計測時点までのGC発生回数を示します。JVMモニタリソースが監視を開始する前に発生したGCの発生回数も値に含みます。
5	半角数字	監視対象Java VMの起動直後から計測時点までのGC総実行時間を示します。単位はミリ秒です。JVMモニタリソースが監視を開始する前に発生したGCの実行時間も値に含みます。

監視対象Java VMの仮想メモリ使用量を確認する(jraruntime.stat)

監視対象Java VMの仮想メモリ使用量を記録するログファイルです。ファイル名はログ出力設定のローテーション方式により以下のいずれかになります。

- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[ファイルサイズ]を選択した場合：jraruntime<*0から始まる整数*>.stat
- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[時間]を選択した場合：jraruntime<*YYYYMMDDhhmm*>.stat

フォーマットは以下の通りです。

No	フォーマット	説明
1	yyyy/mm/dd hh:mm:ss.SSS	ログを記録した日時を示します。
2	半角英数字記号	監視対象Java VMの名称を示します。JVMモニタリソースの[プロパティ]-[監視固有]タブ-[識別名]で設定した値です。
3	半角数字	監視対象Java VMの起動時間をミリ秒単位で示します。
4	半角数字	監視対象Java VMの稼働時間をミリ秒単位で示します。
5	半角英数字記号	監視対象Java VMの仮想メモリ使用量を示します。仮想メモリ使用量にはヒープ領域、非ヒープ領域、スタック領域の使用領域も含まれます。 [プロセス名:PID,使用量:]. 単位はバイトです。

WebLogic Serverのワークマネージャの稼働状況を確認する(wlworkmanager.stat)

WebLogic Serverのワークマネージャの稼働状況を記録するログファイルです。ファイル名はログ出力設定のローテーション方式により以下のいずれかになります。

- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[ファイルサイズ]を選択した場合:wlworkmanager<*0から始まる整数*>.stat
- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[時間]を選択した場合:wlworkmanager<*YYYYMMDDhhmm*>.stat

フォーマットは以下の通りです。

No	フォーマット	説明
1	yyyy/mm/dd hh:mm:ss.SSS	ログを記録した日時を示します。
2	半角英数字記号	監視対象Java VMの名称を示します。JVMモニタリソースの[プロパティ]-[監視固有]タブ-[識別名]で設定した値です。
3	半角英数字記号	アプリケーション名を示します。
4	半角英数字記号	ワークマネージャ名を示します。
5	半角数字	実行したリクエストの数を示します。
6	半角数字	待機しているリクエストの数を示します。

WebLogic Serverのスレッドプールの稼働状況を確認する(wlthreadpool.stat)

WebLogic Serverのスレッドプールの稼働状況を記録するログファイルです。ファイル名はログ出力設定のローテーション方式により以下のいずれかになります。

- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[ファイルサイズ]を選択した場合:wlthreadpool<*0から始まる整数*>.stat
- クラスタプロパティ-[JVM監視]タブ-[ログ出力設定]-[ローテーション方式]-[時間]を選択した場合:wlthreadpool<*YYYYMMDDhhmm*>.stat

フォーマットは以下の通りです。

No	フォーマット	説明
1	yyyy/mm/dd hh:mm:ss.SSS	ログを記録した日時を示します。
2	半角英数字記号	監視対象Java VMの名称を示します。JVMモニタリソースの[プロパティ]-[監視固有]タブ-[識別名]で設定した値です。
3	半角数字	実行したリクエストの総数を示します。
4	半角数字	処理待ちとなっているリクエスト数を示します。
5	半角数字	単位時間(秒)あたりのリクエスト処理数を示します。
6	半角数字	アプリケーションを実行するためのスレッドのトータル数を示します。
7	半角数字	アイドル状態となっているスレッドの数を示します。
8	半角数字	実行中のスレッド数を示します。
9	半角数字	スタンバイ状態となっているスレッド数を示します。

Javaメモリプール名について

JVM運用ログに出力するメッセージ中のmemory_nameとして出力するJavaメモリプール名、およびJVM統計ログjramemory.stat中に出力するJavaメモリプール名について説明します。

Javaメモリプール名として出力する文字列は、JVMモニタリソースで決定しているのではなく、監視対象Java VMから受け取った文字列を出力しています。
また、Java VMとしては仕様を公開していないため、Java VMのバージョンアップにより、予告なく変更される可能性があります。
そのため、メッセージ中のJavaメモリプール名をメッセージ監視することは推奨いたしません。

下記の監視項目とはJVMモニタリソースの[プロパティ]-[監視(固有)] タブ-[メモリ]タブ内の設定項目を表します。
以下に記載しているJavaメモリプール名はOracle Java 、JRockitにおいて実機確認した結果です。

監視対象Java VMの起動オプションに「-XX:+UseSerialGC」が付加されている場合、jramemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列
[ヒープ使用率を監視する]-[領域全体]	HEAP
[ヒープ使用率を監視する]-[Eden Space]	Eden Space
[ヒープ使用率を監視する]-[Survivor Space]	Survivor Space
[ヒープ使用率を監視する]-[Tenured Gen]	Tenured Gen
[非ヒープ使用率を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用率を監視する]-[Code Cache]	Code Cache
[非ヒープ使用率を監視する]-[Perm Gen]	Perm Gen
[非ヒープ使用率を監視する]-[Perm Gen[shared-ro]]	Perm Gen [shared-ro]
[非ヒープ使用率を監視する]-[Perm Gen[shared-rw]]	Perm Gen [shared-rw]

監視対象Java VMの起動オプションに「-XX:+UseParallelGC」、「-XX:+UseParallelOldGC」が付加されている場合、jramemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列
[ヒープ使用率を監視する]-[領域全体]	HEAP
[ヒープ使用率を監視する]-[Eden Space]	PS Eden Space
[ヒープ使用率を監視する]-[Survivor Space]	PS Survivor Space
[ヒープ使用率を監視する]-[Tenured Gen]	PS Old Gen
[非ヒープ使用率を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用率を監視する]-[Code Cache]	Code Cache
[非ヒープ使用率を監視する]-[Perm Gen]	PS Perm Gen
[非ヒープ使用率を監視する]-[Perm Gen[shared-ro]]	Perm Gen [shared-ro]
[非ヒープ使用率を監視する]-[Perm Gen[shared-rw]]	Perm Gen [shared-rw]

監視対象Java VMの起動オプションに「-XX:+UseConcMarkSweepGC」が付加されている場合、jramemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列
[ヒープ使用率を監視する]-[領域全体]	HEAP
[ヒープ使用率を監視する]-[Eden Space]	Par Eden Space

[ヒープ使用率を監視する]-[Survivor Space]	Par Survivor Space
[ヒープ使用率を監視する]-[Tenured Gen]	CMS Old Gen
[非ヒープ使用率を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用率を監視する]-[Code Cache]	Code Cache
[非ヒープ使用率を監視する]-[Perm Gen]	CMS Perm Gen
[非ヒープ使用率を監視する]-[Perm Gen[shared-ro]]	Perm Gen [shared-ro]
[非ヒープ使用率を監視する]-[Perm Gen[shared-rw]]	Perm Gen [shared-rw]

監視対象Java VMがOracle JRockitの場合([JVM 種別]で[JRockit]選択時)、jramemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列
[ヒープ使用率を監視する]-[領域全体]	HEAP memory
[ヒープ使用率を監視する]-[Nursery Space]	Nursery
[ヒープ使用率を監視する]-[Old Space]	Old Space
[非ヒープ使用率を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用率を監視する]-[Class Memory]	Class Memory

[JVM種別]に[Oracle Java(usage monitoring)]を選択、かつ監視対象Java VMの起動オプションに「-XX:+UseSerialGC」が付加されている場合、jramemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列
[ヒープ使用量を監視する]-[領域全体]	HEAP
[ヒープ使用量を監視する]-[Eden Space]	Eden Space
[ヒープ使用量を監視する]-[Survivor Space]	Survivor Space
[ヒープ使用量を監視する]-[Tenured Gen]	Tenured Gen
[非ヒープ使用量を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用量を監視する]-[Code Cache]	Code Cache(Java 9の場合、出力なし)
[非ヒープ使用量を監視する]-[Metaspace]	Metaspace

[JVM種別]に[Oracle Java(usage monitoring)]を選択、かつ監視対象Java VMの起動オプションに「-XX:+UseParallelGC」、「-XX:+UseParallelOldGC」が付加されている場合、jramemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列
[ヒープ使用量を監視する]-[領域全体]	HEAP
[ヒープ使用量を監視する]-[Eden Space]	PS Eden Space
[ヒープ使用量を監視する]-[Survivor Space]	PS Survivor Space
[ヒープ使用量を監視する]-[Tenured Gen]	PS Old Gen
[非ヒープ使用量を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用量を監視する]-[Code Cache]	Code Cache(Java 9の場合、出力なし)
[非ヒープ使用量を監視する]-[Metaspace]	Metaspace

[JVM種別]に[Oracle Java(usage monitoring)]を選択、かつ監視対象Java VMの起動オプションに「-XX:+UseConcMarkSweepGC」が付加されている場合、jramemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列

[ヒープ使用量を監視する]-[領域全体]	HEAP
[ヒープ使用量を監視する]-[Eden Space]	Par Eden Space
[ヒープ使用量を監視する]-[Survivor Space]	Par Survivor Space
[ヒープ使用量を監視する]-[Tenured Gen]	CMS Old Gen
[非ヒープ使用量を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用量を監視する]-[Code Cache]	Code Cache(Java 9の場合、出力なし)
[非ヒープ使用量を監視する]-[Metaspace]	Metaspace

[JVM種別]に[Oracle Java(usage monitoring)]を選択、かつ監視対象Java VMの起動オプションに「-XX:+UseParNewGC」が付加されている場合、jrammemory.statにおけるNo3のJavaメモリプール名は以下の通りです。Java 9の場合、「-XX:+UseParNewGC」を付加すると、監視対象Java VMは起動しません。

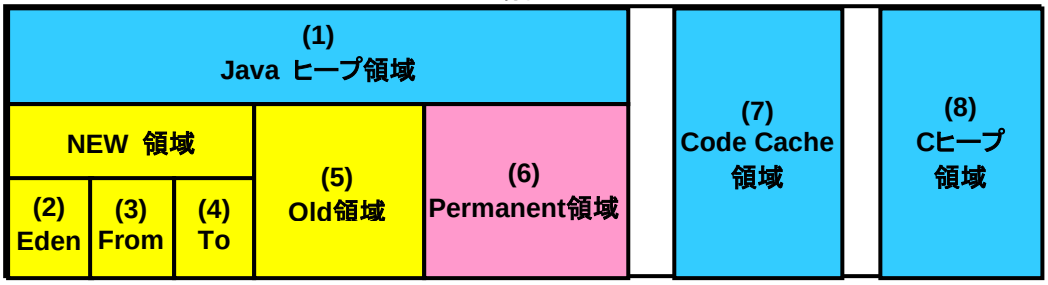
監視項目	memory_nameとして出力する文字列
[ヒープ使用量を監視する]-[領域全体]	HEAP
[ヒープ使用量を監視する]-[Eden Space]	Par Eden Space
[ヒープ使用量を監視する]-[Survivor Space]	Par Survivor Space
[ヒープ使用量を監視する]-[Tenured Gen]	Tenured Gen
[非ヒープ使用量を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用量を監視する]-[Code Cache]	Code Cache
[非ヒープ使用量を監視する]-[Metaspace]	Metaspace

[JVM種別]に[Oracle Java(usage monitoring)]を選択、かつ監視対象Java VMの起動オプションに「-XX::+UseG1GC」が付加されている場合、jrammemory.statにおけるNo3のJavaメモリプール名は以下の通りです。

監視項目	memory_nameとして出力する文字列
[ヒープ使用量を監視する]-[領域全体]	HEAP
[ヒープ使用量を監視する]-[Eden Space]	G1 Eden Space
[ヒープ使用量を監視する]-[Survivor Space]	G1 Survivor Space
[ヒープ使用量を監視する]-[Tenured Gen(Old Gen)]	G1 Old Gen
[非ヒープ使用量を監視する]-[領域全体]	NON_HEAP
[非ヒープ使用量を監視する]-[Code Cache]	Code Cache(Java 9の場合、出力なし)
[非ヒープ使用量を監視する]-[Metaspace]	Metaspace

JVM統計ログjrammemory.statにおけるJavaメモリプール名と、Java VMメモリ空間の関係は以下の通りです。

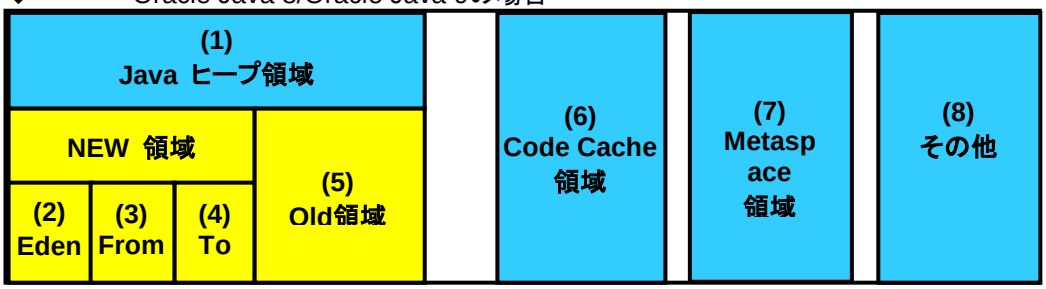
◆ Oracle Java 6/Oracle Java 7の場合



図中の	監視項目	jrammemory.statのJavaメモリプール名

No		
(1)	[ヒープ使用率を監視する]-[領域全体]	HEAP
(2)	[ヒープ使用率を監視する]-[Eden Space]	EdenSpace PS Eden Space Par Eden Space
(3)+(4)	[ヒープ使用率を監視する]-[Survivor Space]	Survivor Space PS Survivor Space Par Survivor Space
(5)	[ヒープ使用率を監視する]-[Tenured Gen]	Tenured Gen PS Old Gen CMS Old Gen
(6)	[非ヒープ使用率を監視する]-[Perm Gen] [非ヒープ使用率を監視する]-[Perm Gen[shared-ro]] [非ヒープ使用率を監視する]-[Perm Gen[shared-rw]]	Perm Gen Perm Gen [shared-ro] Perm Gen [shared-rw] PS Perm Gen CMS Perm Gen
(7)	[非ヒープ使用率を監視する]-[Code Cache]	Code Cache
(8)	ー	ー
(6)+(7)	[非ヒープ使用率を監視する]-[領域全体]	NON_HEAP ※スタックトレースは含みません

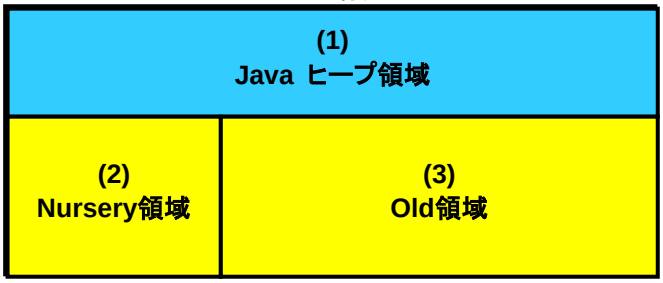
◆ Oracle Java 8/Oracle Java 9の場合



図中の No	監視項目	jramemory.statのJavaメモリプール名
(1)	[ヒープ使用量を監視する]-[領域全体]	HEAP
(2)	[ヒープ使用量を監視する]-[Eden Space]	EdenSpace PS Eden Space Par Eden Space G1 Eden Space
(3)+(4)	[ヒープ使用量を監視する]-[Survivor Space]	Survivor Space PS Survivor Space Par Survivor Space G1 Survivor Space
(5)	[ヒープ使用量を監視する]-[Tenured Gen]	Tenured Gen PS Old Gen CMS Old Gen G1 Old Gen
(6)	[非ヒープ使用量を監視する]-[Code Cache]	Code Cache(Java 9の場合、出力なし)
(7)	[非ヒープ使用量を監視す	Metaspace

	る]-[Metaspace]	
(8)	ー	Compressed Class Spaceなど
(6)+(7)+(8)	[非ヒープ使用量を監視する]-[領域全体]	NON_HEAP

◆ Oracle JRockitの場合



図中のNo	監視項目	jramemory.statのJavaメモリプール名
(1)	[ヒープ使用率を監視する]-[領域全体]	HEAP memory
(2)	[ヒープ使用率を監視する]-[Nursery Space]	Nursery
(3) (注意)	[ヒープ使用率を監視する]-[Old Space]	Old Space
ー	[非ヒープ使用率を監視する]-[領域全体]	NON_HEAP
ー	[非ヒープ使用率を監視する]-[Class Memory]	Class Memory

(注意)
jramemory.stat のJavaメモリプール名"Old Space"については、HEAP内のold領域の値ではなく、"HEAP memory"全体と同値となります。(3)のみの計測はできません。

異常検出時に障害原因別にコマンドを実行するには

モニタリソースの異常検出時、CLUSTERPROでは障害原因別に異なるコマンドを区別して実行する手段を提供していません。
JVMモニタリソースでは障害原因別にコマンドを区別して実行可能です。異常検出時に実行します。

障害原因別に実行するコマンドの設定項目は以下の通りです。

障害原因	設定項目
・監視対象のJava VMへ接続失敗 ・リソース計測失敗	[監視(固有)]タブ-[コマンド]
・ヒープ使用率 ・非ヒープ使用率 ・仮想メモリ使用量 ・ヒープ使用量 ・非ヒープ使用量	[監視(固有)]タブ-[調整]プロパティ-[メモリ]タブ-[コマンド]
・動作中のスレッド数	[監視(固有)]タブ-[調整]プロパティ-[スレッド]タブ-[コマンド]
・Full GC実行時間 ・Full GC発生回数	[監視(固有)]タブ-[調整]プロパティ-[GC]タブ-[コマンド]

・WebLogic のワークマネージャのリクエスト ・WebLogic のスレッドプールのリクエスト	[監視(固有)]タブ-[調整]プロパティ-[WebLogic]タブ-[コマンド]

[コマンド]は障害原因の詳細をコマンドの引数として渡します。引数は[コマンド]の最後に結合して渡します。スクリプトなどを自身で作成し[コマンド]へ設定することにより、更に障害原因に特化した動作が可能です。引数として渡す文字列は以下の通りです。
引数として渡す文字列が複数記載している場合は、監視対象Java VMのGC方式によりいずれかを渡します。差異の詳細は「Javaメモリプール名について」を参照してください。
(Oracle Javaの場合) (Oracle JRockitの場合)と記載がある場合は、JVM種別により異なります。記載がない場合、JVM種別による区別はありません。

障害原因の詳細	引数として渡す文字列
・監視対象のJava VMへ接続失敗 ・リソース計測失敗	なし
[監視(固有)]タブ-[調整]プロパティ-[メモリ]タブ-[ヒープ使用率を監視する]-[領域全体] (Oracle Javaの場合)	HEAP
[メモリ]タブ- [ヒープ使用率を監視する]-[Eden Space] (Oracle Javaの場合)	EdenSpace PSEdenSpace ParEdenSpace
[メモリ]タブ- [ヒープ使用率を監視する]-[Survivor Space] (Oracle Javaの場合)	SurvivorSpace PSSurvivorSpace ParSurvivorSpace
[メモリ]タブ- [ヒープ使用率を監視する]-[Tenured Gen] (Oracle Javaの場合)	TenuredGen PSOldGen CMSOldGen
[メモリ]タブ- [非ヒープ使用率を監視する]-[領域全体] (Oracle Javaの場合)	NON_HEAP
[メモリ]タブ- [非ヒープ使用率を監視する]-[Code Cache] (Oracle Javaの場合)	CodeCache
[メモリ]タブ- [非ヒープ使用率を監視する]-[Perm Gen] (Oracle Javaの場合)	PermGen PSPermGen CMSPermGen
[メモリ]タブ- [非ヒープ使用率を監視する]-[Perm Gen[shared-ro]] (Oracle Javaの場合)	PermGen[shared-ro]
[メモリ]タブ- [非ヒープ使用率を監視する]-[Perm Gen[shared-rw]] (Oracle Javaの場合)	PermGen[shared-rw]
[メモリ]タブ- [仮想メモリ使用量を監視する]- (Oracle Javaの場合)	Vmsize
[メモリ]タブ- [ヒープ使用量を監視する]-[領域全体](Oracle Java(usage monitoring)の場合)	HEAP
[メモリ]タブ- [ヒープ使用量を監視する]-[Eden Space](Oracle Java(usage monitoring)の場合)	EdenSpace PSEdenSpace ParEdenSpace G1EdenSpace
[メモリ]タブ- [ヒープ使用量を監視する]-[Survivor Space](Oracle Java(usage monitoring)の場合)	SurvivorSpace PSSurvivorSpace ParSurvivorSpace

	G1SurvivorSpace
[メモリ]タブ- [ヒープ使用量を監視する]-[Tenured Gen](Oracle Java(usage monitoring)の場合)	TenuredGen PSOldGen CMSOldGen G1OldGen
[メモリ]タブ- [非ヒープ使用量を監視する]-[領域全体](Oracle Java(usage monitoring)の場合)	NON_HEAP
[メモリ]タブ- [非ヒープ使用量を監視する]-[Code Cache](Oracle Java(usage monitoring)の場合)	CodeCache
[メモリ]タブ- [非ヒープ使用量を監視する]-[Metaspace](Oracle Java(usage monitoring)の場合)	Metaspace
[メモリ]タブ- [ヒープ使用率を監視する]-[領域全体] (Oracle JRockitの場合)	HEAP Heap
[メモリ]タブ- [ヒープ使用率を監視する]-[Nursery Space] (Oracle JRockitの場合)	Nursery
[メモリ]タブ- [ヒープ使用率を監視する]-[Old Space] (Oracle JRockitの場合)	OldSpace
[メモリ]タブ- [非ヒープ使用率を監視する]-[領域全体] (Oracle JRockitの場合)	NON_HEAP
[メモリ]タブ- [非ヒープ使用率を監視する]-[Class Memory] (Oracle JRockitの場合)	ClassMemory
[メモリ]タブ- [仮想メモリ使用量を監視する]- (Oracle JRockitの場合)	Vmsize
[スレッド]タブ-[動作中のスレッド数を監視する]	Count
[GC]タブ-[Full GC 実行時間を監視する]	Time
[GC]タブ-[Full GC 発生回数を監視する]	Count
[WebLogic]タブ-[ワークマネージャのリクエストを監視する]-[待機リクエスト リクエスト数]	WorkManager_PendingRequests
[WebLogic]タブ-[スレッドプールのリクエストを監視する]-[待機リクエスト リクエスト数]	ThreadPool_PendingUserRequestCount
[WebLogic]タブ-[スレッドプールのリクエストを監視する]-[実行リクエスト リクエスト数]	ThreadPool_Throughput

以下に実行例に示します。
例1)

設定項目	設定内容
[監視(固有)]タブ-[調整]プロパティ-[GC]タブ-[コマンド]	/usr/local/bin/downcmd
[監視(固有)]タブ-[調整]プロパティ-[GC]タブ-[Full GC 発生回数を監視する]	1
[クラスタ]プロパティ-[JVM監視]タブ-[リソース計測設定]-[共通]タブ-[異常判定しきい値	3

JVMモニタリソースは、異常判定しきい値回(3回)連続してFull GCが発生すると、モニタ異常を検出し、「/usr/local/bin/downcmd Cont」としてコマンドを実行します。

例2)

設定項目	設定内容
[監視(固有)]タブ-[調整]プロパティ-[GC]タブ-[コマンド]	“/usr/local/bin/downcmd” GC

[監視(固有)]タブ-[調整]プロパティ-[GC]タブ-[Full GC実行時間を監視する]	65536
[クラスタ]プロパティ-[JVM監視]タブ-[リソース計測設定]-[共通]タブ-[異常判定しきい値	3

JVMモニタリソースは、異常判定しきい値回(3回)連続してFull GC 実行時間が 65535 ミリ秒超過すると、モニタ異常を検出し、「/usr/local/bin/downcmd GC Time」としてコマンドを実行します。

例3)

設定項目	設定内容
[監視(固有)]タブ-[調整]プロパティ-[メモリ]タブ-[コマンド]	“/usr/local/bin/downcmd” memory
[監視(固有)]タブ-[調整]プロパティ-[メモリ]タブ-[ヒープ使用率を監視する]	オン
[監視(固有)]タブ-[調整]プロパティ-[メモリ]タブ-[Eden Space]	80
[監視(固有)]タブ-[調整]プロパティ-[メモリ]タブ-[Survivor Space]	80
[クラスタ]プロパティ-[JVM監視]タブ-[リソース計測設定]-[共通]タブ-[異常判定しきい値]	3

JVMモニタリソースは、異常判定しきい値回(3回)連続してJava Eden Space の使用率および Java Survivor Space の使用率が 80% を超過すると、モニタ異常を検出し、「/usr/local/bin/downcmd memory EdenSpace SurvivorSpace」としてコマンドを実行します。

[コマンド]で設定したコマンドの終了を待つタイムアウト(秒)は、クラスタプロパティ-[JVM監視]タブ-[コマンドタイムアウト]で設定します。これは上記各タブの[コマンド]で同じ値を適用します。[コマンド]個別には設定できません。
タイムアウトした場合、[コマンド]プロセスを強制終了させるような処理は実行しません。[コマンド]プロセスの後処理(例:強制終了)は、お客様が実行してください。タイムアウトした場合は、以下のメッセージをJVM運用ログへ出力します。

action thread execution did not finish. action is alive = <*コマンド*>

注意事項は以下の通りです。

- Java VM の正常復帰検出時(異常→正常時)には[コマンド]は実行しません。
- [コマンド]は Java VM 異常検出時(しきい値の超過が異常判定しきい値回連続して発生した場合)を契機として実行します。しきい値の超過毎には実行しません。
- 複数のタブにて[コマンド]を設定すると、同時に障害が発生した場合は複数の[コマンド]が実行されます。そのため、システム負荷にはご注意ください。
- [監視(固有)]タブ-[調整]プロパティ-[WebLogic]タブ-[ワークマネージャのリクエストを監視する]-[待機リクエスト リクエスト数]、[監視(固有)]タブ-[調整]プロパティ-[WebLogic]タブ-[ワークマネージャのリクエストを監視する]-[待機リクエスト 平均値]を両方監視している場合、[コマンド]が同時に 2 回実行される可能性があります。
  これは、[クラスタ]プロパティ-[JVM 監視]タブ-[リソース計測設定]-[WebLogic]タブ-[インターバル リクエスト数]と[クラスタ]プロパティ-[JVM 監視]タブ-[リソース計測設定]-[WebLogic]タブ-[インターバル 平均値]の異常検出が同時に発生する可能性があるためです。回避策としては、どちらか一方のみ監視するようにしてください。以下の監視項目の組み合わせも同様です。

- [監視(固有)]タブ-[調整]プロパティ-[WebLogic]タブ-[スレッドプールのリクエストを監視する]-[待機リクエスト リクエスト数]と、[監視(固有)]タブ-[調整]プロパティ-[WebLogic]タブ-[スレッドプールのリクエストを監視する]-[待機リクエスト 平均値]
- [監視(固有)]タブ-[調整]プロパティ-[WebLogic]タブ-[スレッドプールのリクエストを監視する]-[実行リクエスト リクエスト数]と、[監視(固有)]タブ-[調整]プロパティ-[WebLogic]タブ-[スレッドプールのリクエストを監視する]-[実行リクエスト 平均値]

WebLogic Serverを監視するには

監視対象のWebLogic Serverの設定が終了しアプリケーションサーバとして稼動させる手順は、WebLogic Serverのマニュアルを参照してください。

本章では、JVM モニタリソースで監視するために必要な設定のみについて記述します。

1. WebLogic Server Administration Console を起動します。

   起動方法は、WebLogic Server マニュアルの「Administration Console の概要」を参照してください。

   ドメインコンフィグレーション－ドメイン－コンフィグレーション－全般を選択します。ここで**「管理ポートの有効化」のチェックがオフ**になっていることを確認してください

2. ドメインコンフィグレーション-サーバを選択し、監視対象のサーバ名を選択します。選択したサーバ名は Builder ツリービューから選択可能な[プロパティ]-[監視(固有)] タブの識別名に設定します。「JVM モニタリソースを理解する」を参照してください。

3. 監視対象のサーバのコンフィグレーション－全般で**「リスンポート」**で管理接続するポート番号を確認します。

4. WebLogic Server を停止します。停止方法は、WebLogic Server マニュアルの「WebLogic Server の起動と停止」を参照してください。

5. WebLogic Server の管理サーバ起動スクリプト(startWebLogic.sh)を開きます。

6. 開いたスクリプトに以下の内容を記述します。

   - 監視対象が WebLogic Server の管理サーバの場合
     - ✓ **JAVA_OPTIONS="${JAVA_OPTIONS}**
       -Dcom.sun.management.jmxremote.port=*n*
       -Dcom.sun.management.jmxremote.ssl=false
       -Dcom.sun.management.jmxremote.authenticate=false"
       ※上記内容は実際には 1 行で記述してください。
   - 監視対象が WebLogic Server の管理対象サーバの場合
     - ✓ **if ["${SERVER_NAME}" = "*SERVER_NAME*"]; then**
       JAVA_OPTIONS="${JAVA_OPTIONS}
       -Dcom.sun.management.jmxremote.port=*n*
       -Dcom.sun.management.jmxremote.ssl=false
       -Dcom.sun.management.jmxremote.authenticate=false"
       fi

       ※上記で if 文の中(2～5 行目)は実際には 1 行で記述してください。

注: ***n*** は、監視のために使用するポート番号を指定します。指定するポート番号は**監視対象の Java VM のリスンポート番号とは別の番号を指定してください。**また同一のマシンに複数の監視対象の WebLogic Server が存在する場合、そのリスニング・ポート番号や他のアプリケーションのポート番号と重複しないポート番号を指定してください。

注: ***SERVER_NAME*** は、「監視対象サーバ選択」で確認した監視対象となるサーバ名を指定します。監視対象のサーバが複数の場合、同様の設定(1～6 行目)に対してサーバ名を変更し、繰り返し設定してください。

注:監視対象が WebLogic Server 11gR1(10.3.3) 以降の場合、以下のオプションも追加してください。

-Djavax.management.builder.initial=weblogic.management.jmx.

mbeanserver.WLSMBeanServerBuilder

注:上記の記述内容の追加箇所は、以下の記述より前に記述するようにしてください。

${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
-Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy
${PROXY_SETTINGS} ${SERVER_CLASS}

※上記内容は実際には 1 行で記述してください

※WebLogic のバージョンによって、上記の java 引数の内容が異なっている場合がありますが、java の実行前に JAVA_OPTIONS を記述していただければ問題ありません。

注:[メモリタブ]の[Perm Gen[shared-ro]]や [Perm Gen[shared-rw]]を監視する場合、

以下を追加してください。

-client –Xshare:on –XX:+UseSerialGC

7. 監視対象の WebLogic Server の標準出力、標準エラー出力をファイルにリダイレクトします。設定方法は、WebLogic Server のマニュアルを参照してください。標準出力、標準エラー出力を情報採取に含めたい場合は設定してください。設定する場合、ハードディスクの容量が不足しないよう注意してください。
8. 監視対象の WebLogic Server に GC ログを出力する設定をします。設定方法は、WebLogic Server のマニュアルを参照してください。GC ログを情報採取に含めたい場合は設定してください。設定する場合、ハードディスクの容量が不足しないよう注意してください。
9. 以下の設定を行ってください。

   監視対象の WebLogic Server の WLST(wlst.sh)を起動します。表示されたコンソール画面上で、以下のコマンドを実行してください。

   >connect('*USERNAME***','***PASSWORD***','t3://***SERVER_ADDRESS:SERVER_PORT***')**
   > edit()
   > startEdit()
   > cd('JMX/*DOMAIN_NAME***')**
   > set('PlatformMBeanServerUsed','true')
   > activate()
   > exit()

   上記の **USERNAME**、**PASSWORD**、**SERVER_ADDRESS**、**SERVER_PORT**、**DOMAIN_NAME** はドメイン環境に応じた値に置き換えてください
10. 監視対象の WebLogic Server を再起動します

WebOTXを監視するには

本書では、JVM モニタリソースで監視する対象の WebOTX の設定手順について記述します。

WebOTX 統合運用管理コンソールを起動します。起動方法は「WebOTX 運用編（Web 版統合運用管理ツール）」マニュアルの「統合運用管理ツールの起動と終了」を参照してください。

以降の設定は、WebOTX上のJMXエージェントのJavaプロセスに対する監視を行う場合と、プロセスグループ上のJavaプロセスに対する監視を行う場合とで設定内容が異なります。監視する対象に合わせて、設定してください。

WebOTX ドメインエージェントのJavaプロセスを監視するには

特に設定作業は不要です。V8.30をご利用の場合は、V8.31以降にアップデートしてください。

WebOTX プロセスグループのJavaプロセスを監視するには

1. 統合運用管理ツールよりドメインと接続します。
2. ツリービューより[<ドメイン名>]-[TP システム]-[アプリケーショングループ]-[<アプリケーショングループ名>]-[プロセスグループ]-[<プロセスグループ名>]を選択します。
3. 右側に表示される[JVM オプション]タブ内の[その他の引数]属性に、次の Java オプションを 1 行で指定します。***n*** は、ポート番号を指定します。同一のマシンに複数の監視対象の Java VM が存在する場合、重複しないポート番号を指定してください。ここで指定するポート番号は、Builder(モニタリソースプロパティ→監視(固有)タブ→接続ポート番号)でも設定します。

   -Dcom.sun.management.jmxremote.port=*n*
   -Dcom.sun.management.jmxremote.ssl=false
   -Dcom.sun.management.jmxremote.authenticate=false
   -Djavax.management.builder.initial=com.nec.webotx.jmx.mbeanserver.JmxMBeanServerBuilder

   ※WebOTX V9.2以降では -Djavax.management.builder.initial の指定は不要です。

4. 設定後、[更新]ボタンを押します。設定が完了したら、プロセスグループを再起動します。

   本設定は、WebOTX 統合運用管理ツールの[Java システムプロパティ]タブ内の[Java システムプロパティ]属性にて指定することも可能です。その場合は、”-D”は指定せず、また、”=”より前の文字列を「名前」に、”=”より後ろの文字列を「値」に指定してください。

注:WebOTX プロセスグループの機能でプロセス障害時の再起動を設定されている場合、CLUSTERPRO からの復旧動作でプロセスグループの再起動を実行すると、WebOTX プロセスグループの機能が正常に動作しない場合があります。そのため、WebOTX プロセスグループを監視する場合は Builder から JVM 監視リソースに対して以下のように設定してください。

設定タブ名	項目名	設定値
監視(共通)	監視タイミング	常時
回復動作	回復動作	最終動作のみ実行
回復動作	最終動作	何もしない

また、WebOTX プロセスグループの監視について、ロードバランサ連携機能はサポートしておりません。

WebOTX notification通知を受信するには

特定のリスナクラスを登録することにより、WebOTXが障害を検出するとnotificationが発行されます。JVM監視リソースはそのnotificationを受信し、JVM運用ログへ以下のメッセージを通知します。
%1$s:Notification received. %2$s.

%1$s、%2$sの意味は以下のとおりです。
%1$s：監視対象Java VM
%2$s：notificationの通知メッセージ (ObjectName=**,type=**,message=**)

現在、監視可能なリソースのMBeanの詳細情報は以下のとおりです。

ObjectName	[domainname]:j2eeType=J2EEDomain,name=[domainname], category=runtime
notificationタイプ	nec.webotx.monitor.alivecheck.not-alive
メッセージ	failed

JBossを監視するには

JBoss Enterprise Application Platform 6 以降、スタンドアローンモードとドメインモードという管理モードがサポートされました。JVM モニタリソースで監視するためには、以下 3 通りの場合分けが必要です。

- JBoss Enterprise Application Platform 5 以前の場合
- JBoss Enterprise Application Platform 6 のスタンドアローンモードの場合
- JBoss Enterprise Application Platform 6 のドメインモードの場合

JVM モニタリソースで監視する対象の JBoss の設定手順について記述します。

JBoss Enterprise Application Platform 5 以前の場合

1. JBoss を停止し、(*JBoss インストールパス*)/bin/run.conf をエディタから開きます。
2. 開いた設定ファイルに以下の内容を 1 行で記述します。***n*** は、ポート番号を指定します。同一のマシンに複数の監視対象の Java VM が存在する場合、重複しないポート番号を指定してください。ここで指定するポート番号は、Builder(モニタリソースプロパティ→監視(固有)タブ→接続ポート番号)でも設定します。

   JAVA_OPTS="${JAVA_OPTS}
   -Dcom.sun.management.jmxremote.port=*n*
   -Dcom.sun.management.jmxremote.ssl=false
   -Dcom.sun.management.jmxremote.authenticate=false”

3. 上記設定を保存した後、JBoss を起動します。
4. Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→識別名) には他の監視対象と重ならない任意の文字列(例：JBoss)を設定してください。また、Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→プロセス名) では「com.sun.management.jmxremote.port=*n*」(*n* は 2 で指定したポート番号)を設定してください。

JBoss Enterprise Application Platform 6 のスタンドアローンモードの場合

1. JBoss を停止し、(*JBoss インストールパス*)/bin/standalone.conf をエディタから開きます。

2. 開いた設定ファイルに以下の内容を記述します。***n*** は、ポート番号を指定します。同一のマシンに複数の監視対象の Java VM が存在する場合、重複しないポート番号を指定してください。ここで指定するポート番号は、Builder(モニタリソースプロパティ→監視(固有)タブ→接続ポート番号)でも設定します。

   「if ["x$JBOSS_MODULES_SYSTEM_PKGS" = "x"]; then」より前に以下を追加
   JBOSS_MODULES_SYSTEM_PKGS="org.jboss.logmanager"

   「if ["x$JAVA_OPTS" = "x"]; then … fi:」より後に以下を追加
   JAVA_OPTS="$JAVA_OPTS
   -Xbootclasspath/p:$JBOSS_HOME/modules/org/jboss/logmanager/main/jboss-logmanager-1.3.2.Final-redhat-1.jar"
   JAVA_OPTS="$JAVA_OPTS
   -Djava.util.logging.manager=org.jboss.logmanager.LogManager"
   JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=n
   -Dcom.sun.management.jmxremote.ssl=false
   -Dcom.sun.management.jmxremote.authenticate=false"

   ※jboss-logmanager-*.jar は JBoss のバージョンによって格納ディレクトリ、ファイル名が異なりますため、インストールしている環境に合わせてパスを指定してください。

3. 上記設定を保存した後、JBoss を起動します。

4. Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→識別名) には他の監視対象と重ならない任意の文字列(例：JBoss)を設定してください。また、Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→プロセス名) では「com.sun.management.jmxremote.port=*n*」(*n* は 2 で指定したポート番号)を設定してください。

JBoss Enterprise Application Platform 6 のドメインモードの場合

1. Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→識別名) には他の監視対象と重ならない任意の文字列(例：JBoss)を設定してください。また、Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→プロセス名) では、ユニークに特定できるよう Java VM 起動時オプションを全て指定してください。

Tomcatを監視するには

JVM モニタリソースで監視する対象の Tomcat の設定手順について記述します。

1. Tomcat を停止し、(*Tomcat インストールパス*)/bin/catalina.sh をエディタから開きます。Tomcat を rpm パッケージでインストールした場合は/etc/sysconfig/tomcat6 または /etc/sysconfig/tomcat を開きます。
2. 開いた設定ファイルの Java オプションに以下の内容を 1 行で記述します。***n*** は、ポート番号を指定します。同一のマシンに複数の監視対象の Java VM が存在する場合、重複しないポート番号を指定してください。ここで指定するポート番号は、Builder(モニタリソースプロパティ→監視(固有)タブ→接続ポート番号)でも設定します。
   CATALINA_OPTS="${CATALINA_OPTS}
   -Dcom.sun.management.jmxremote.port=*n*
   -Dcom.sun.management.jmxremote.ssl=false
   -Dcom.sun.management.jmxremote.authenticate=false"

注:catalina.sh の場合、上記記述内容の追加箇所は、以下の記述より前に記述するようにしてください。

```
if [ "$1" = "debug" ] ; then
    if $os400; then
        echo "Debug command not available on OS400"
        exit 1
    else
```

---

3. 上記設定を保存した後、Tomcat を起動します。
4. Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→識別名) には他の監視対象と重ならない任意の文字列(例:tomcat)を設定してください。また、Builder(テーブルビュー→JVM モニタリソース名→[プロパティ]→[監視(固有)] タブ→プロセス名) では「com.sun.management.jmxremote.port=*n*」(*n* は 2 で指定したポート番号)を設定してください。

SVFを監視するには

JVM モニタリソースで監視する対象の SVF の設定手順について記述します。

1. 監視対象を下記より選択し、該当するスクリプトをエディタから開きます。

監視対象	編集するスクリプト
Simple Httpd Service(8.xの場合)	<SVFインストールパス>/bin/SimpleHttpd
Simple Httpd Service(9.xの場合)	<SVFインストールパス>/bin/UCXServer
RDE Service	<SVFインストールパス >/rdjava/rdserver/rd_server_startup.sh
	<SVFインストールパス >/rdjava/rdserver/svf_server_startup.sh
RD Spool Balancer	<SVFインストールパス >/rdjava/rdbalancer/rd_balancer_startup.sh
Tomcat(8.xの場合)	<SVFインストールパス >/rdjava/apache-tomcat-5.5.25/bin/catalina.sh
Tomcat(9.xの場合)	<SVFインストールパス >/apache-tomcat/bin/catalina.sh
SVF Print Spooler Service	<SVFインストールパス>/bin/spooler

2. Java オプション指定箇所に下記の内容を 1 行で記述します。***n*** は、ポート番号を指定します。同一のマシンに複数の監視対象の Java VM が存在する場合、重複しないポート番号を指定してください。ここで指定するポート番号は、Builder(モニタリソースプロパティ→監視(固有)タブ→接続ポート番号)でも設定します。
JAVA_OPTIONS="${JAVA_OPTIONS}
-Dcom.sun.management.jmxremote.port=*n*
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false"

3. 監視対象がRDE Service の場合、下記の起動パス中および rd_balancer_startup.sh に${JAVA_OPTIONS}を追記します。

   java -Xmx256m -Xms256m -Djava.awt.headless=true ${JAVA_OPTIONS} -classpath $CLASSPATH jp.co.fit.vfreport.RdSpoolPlayerServer &

4. Builder(モニタリソースプロパティ→監視(固有)タブ→識別名、Builder(モニタリソースプロパティ→監視(固有)タブ→プロセス名は下記を指定してください。

監視対象	識別名、プロセス名
Simple Httpd Service	SimpleHttpd
RDE Service	ReportDirectorServer
	RdSpoolPlayerServer
RD Spool Balancer	ReportDirectorSpoolBalancer
Tomcat(8.xの場合)	Bootstrap
Tomcat(9.xの場合)	-Dcom.sun.management.jmxremote.port=*n*
SVF Print Spooler Service	spooler.Daemon

iPlanet Web Serverを監視するには

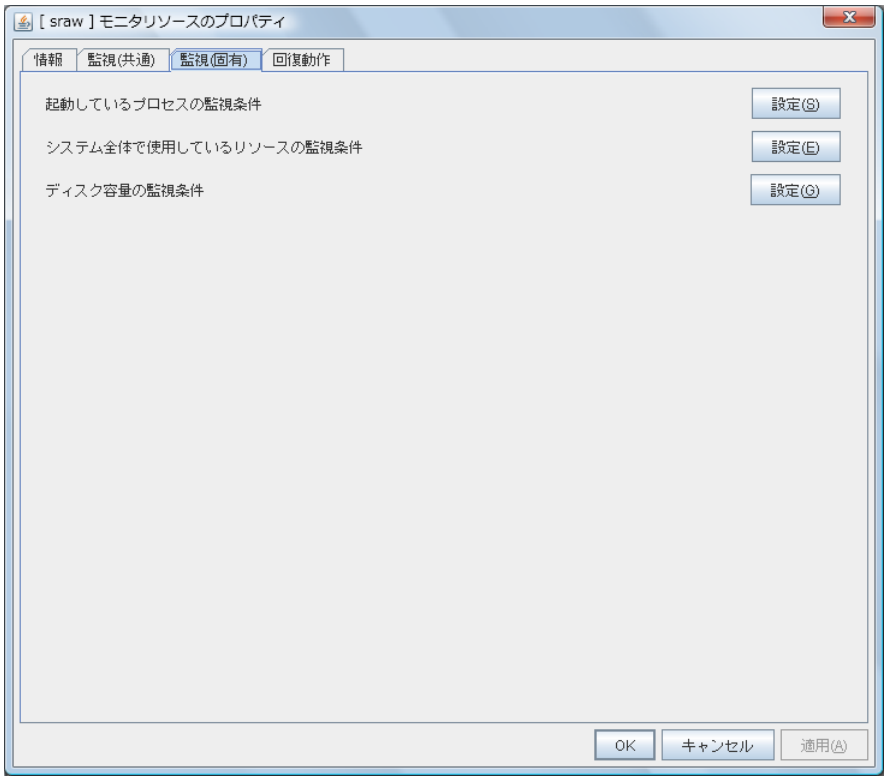
JVM モニタリソースで監視する対象の iPlanet Web Server の設定手順について記述します。

1. iPlanet Web Server を停止し、(iPlanet Web Server インストールパス)/(監視対象サーバ名)/config/server.xml をエディタから開きます。
2. /server/jvm/jvm-options に下記を以下の内容を 1 行で記述します。***n*** は、ポート番号を指定します。同一のマシンに複数の監視対象の Java VM が存在する場合、重複しないポート番号を指定してください。ここで指定するポート番号は、Builder(モニタリソースプロパティ→監視(固有)タブ→接続ポート番号)でも設定します。
   <java-options> -Dcom.sun.management.jmxremote.port=*n*
   -Dcom.sun.management.jmxremote.ssl=false
   -Dcom.sun.management.jmxremote.authenticate=false <java-options>
3. 上記設定を保存した後、iPlanet Web Server を起動します。

システムモニタリソースの設定

システムモニタリソースは、プロセスが使用するリソースの統計情報を継続的に収集し、一定のナレッジ情報にしたがい解析を行います。解析の結果からリソース枯渇の発生を早期検出する機能を提供します。

1. Builder 左部分に表示されているツリービューで、[Monitors] のアイコンをクリックします。
2. 画面右のテーブルビューに、モニタリソースの一覧が表示されます。目的のシステムモニタリソース名を右クリックし、[プロパティ] の [監視(固有)] タブをクリックします。
3. [監視(固有)] タブで、以下の説明に従い詳細設定の表示 / 変更を行います。



[設定]

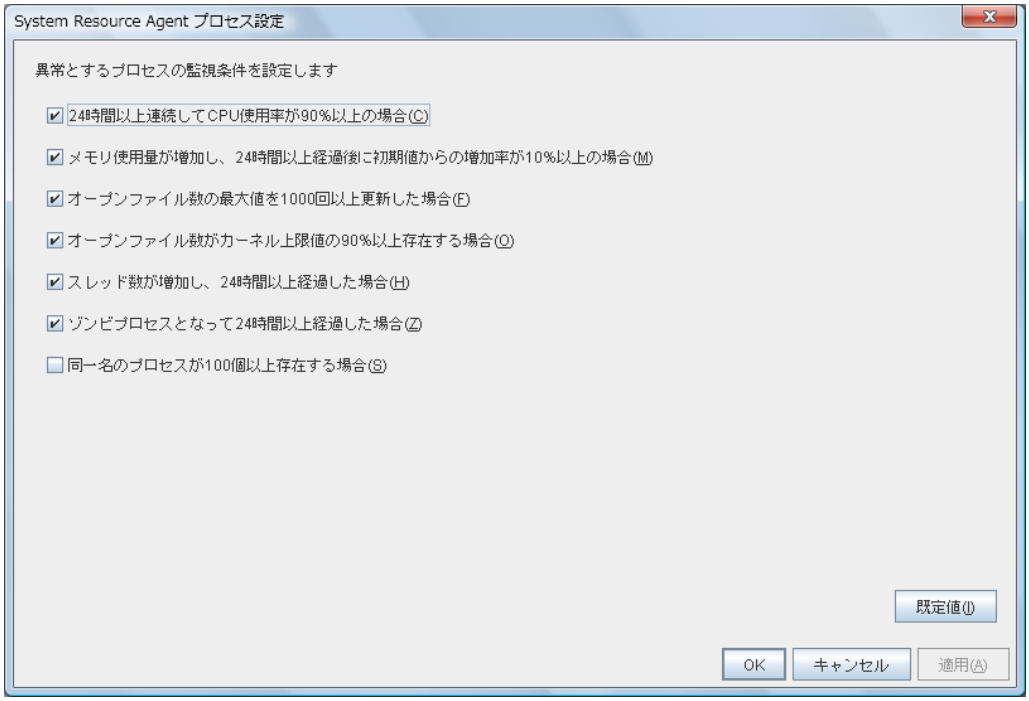
起動しているプロセスの監視条件の[設定]ボタンを選択するとプロセス設定ダイアログが表示されます。

システム全体で使用しているリソースの監視条件の[設定]ボタンを選択するとシステム設定ダイアログが表示されます。

ディスク容量の監視条件の[設定]ボタンを選択するとディスク一覧ダイアログが表示されます。

それぞれのダイアログの説明に従い異常とする監視条件の詳細設定を行います。

System Resource Agent プロセス設定



24 時間以上連続して CPU 使用率が 90%以上の場合

24 時間以上連続して CPU 使用率が 90%以上であるプロセスの監視を行うかどうかを設定します。

- チェックボックスがオン

24 時間以上連続して CPU 使用率が 90%以上であるプロセスの監視を行います。

- チェックボックスがオフ

24 時間以上連続して CPU 使用率が 90%以上であるプロセスの監視を行いません。

メモリ使用量が増加し、24 時間以上経過後に初期値からの増加率が 10%以上の場合

メモリ使用量が増加し、24 時間以上経過後に初期値からの増加率が 10%以上であるプロセスの監視を行うかどうかを設定します。

- チェックボックスがオン

メモリ使用量が増加し、24 時間以上経過後に初期値からの増加率が 10%以上であるプロセスの監視を行います。

- チェックボックスがオフ

メモリ使用量が増加し、24 時間以上経過後に初期値からの増加率が 10%以上であるプロセスの監視を行いません。

オープンファイル数の最大値を 1000 回以上更新した場合

オープンファイル数の最大値を 1000 回以上更新したプロセスの監視を行うかどうかを設定します。

- チェックボックスがオン

オープンファイル数の最大値を 1000 回以上更新したプロセスの監視を行います。

- チェックボックスがオフ

オープンファイル数の最大値を 1000 回以上更新したプロセスの監視を行いません。

オープンファイル数がカーネル上限値の 90%以上存在する場合

オープンファイル数がカーネル上限値の 90%以上存在するプロセスの監視を行うかどうかを設定します。

- チェックボックスがオン

オープンファイル数がカーネル上限値の 90%以上存在するプロセスの監視を行います。

- チェックボックスがオフ

オープンファイル数がカーネル上限値の 90%以上存在するプロセスの監視を行いません。

スレッド数が増加し、24 時間以上経過した場合

スレッド数が増加し、24 時間以上経過したプロセスの監視を行うかどうかを設定します。

- チェックボックスがオン

スレッド数が増加し、24 時間以上経過したプロセスの監視を行います。

- チェックボックスがオフ

スレッド数が増加し、24 時間以上経過したプロセスの監視を行いません。

ゾンビプロセスとなって 24 時間以上経過した場合

ゾンビプロセスとなって 24 時間以上経過したプロセスの監視を行うかどうかを設定します。

- チェックボックスがオン

ゾンビプロセスとなって 24 時間以上経過したプロセスの監視を行います。

- チェックボックスがオフ

ゾンビプロセスとなって 24 時間以上経過したプロセスの監視を行いません。

同一名のプロセスが 100 個以上存在する場合

同一名のプロセスが 100 個以上存在するプロセスの監視を行うかどうかを設定します。

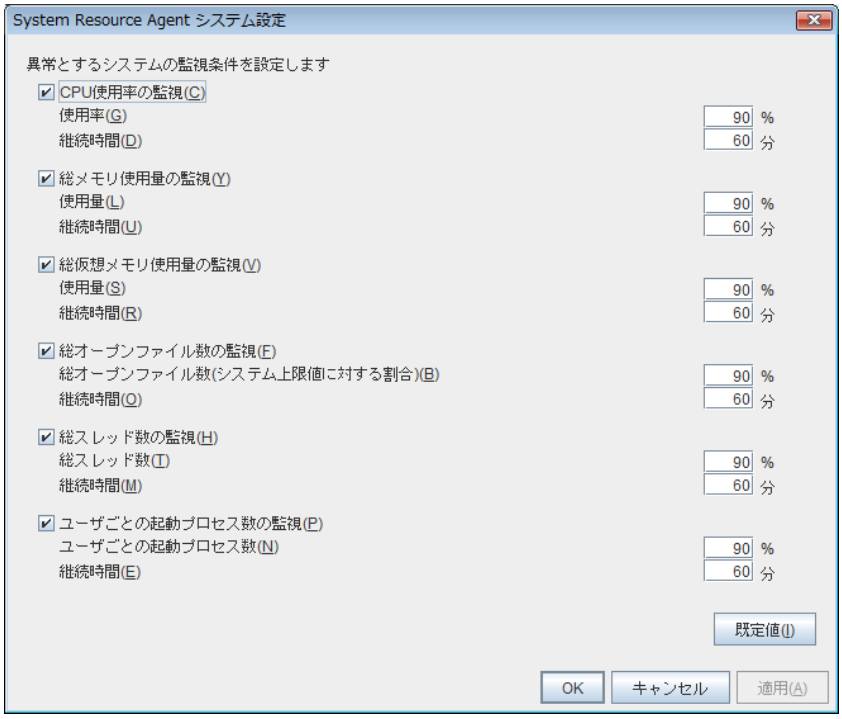
- チェックボックスがオン

同一名のプロセスが 100 個以上存在するプロセスの監視を行います。

- チェックボックスがオフ

同一名のプロセスが 100 個以上存在するプロセスの監視を行いません。

System Resource Agent システム設定



CPU 使用率の監視

CPU 使用率の監視を行うかどうかを設定します。

- チェックボックスがオン

CPU 使用率の監視を行います。

- チェックボックスがオフ

CPU 使用率の監視を行いません。

CPU 使用率 (1～100)

CPU 使用率の異常を検出するしきい値を設定します。

継続時間 (1～1440)

CPU 使用率の異常を検出する時間を設定します。

指定した時間以上連続してしきい値を超過した場合、異常を検出します。

総メモリ使用量の監視

総メモリ使用量の監視を行うかどうかを設定します。

- チェックボックスがオン

総メモリ使用量の監視を行います。

- チェックボックスがオフ

総メモリ使用量の監視を行いません。

総メモリ使用量 (1～100)

メモリの使用量の異常を検出するしきい値(システムのメモリ搭載量に対する割合)を設定します。

継続時間 (1～1440)

総メモリ使用量の異常を検出する時間を設定します。

指定した時間以上連続してしきい値を超過した場合、異常を検出します。

総仮想メモリ使用量の監視

総仮想メモリ使用量の監視を行うかどうかを設定します。

- チェックボックスがオン

総仮想メモリ使用量の監視を行います。

- チェックボックスがオフ

総仮想メモリ使用量の監視を行いません。

総仮想メモリ使用量 (1～100)

仮想メモリの使用量の異常を検出するしきい値を設定します。

継続時間 (1～1440)

総仮想メモリ使用量の異常を検出する時間を設定します。
指定した時間以上連続してしきい値を超過した場合、異常を検出します。

総オープンファイル数の監視

総オープンファイル数の監視を行うかどうかを設定します。

- チェックボックスがオン

総オープンファイル数の監視を行います。

- チェックボックスがオフ

総オープンファイル数の監視を行いません。

総オープンファイル数 (1～100)

オープンしているファイルの総数の異常を検出するしきい値(システム上限値に対する割合)を設定します。

継続時間 (1～1440)

総オープンファイル数の異常を検出する時間を設定します。

指定した時間以上連続してしきい値を超過した場合、異常を検出します。

総スレッド数の監視

総スレッド数の監視を行うかどうかを設定します。

- チェックボックスがオン

総スレッド数の監視を行います。

- チェックボックスがオフ

総スレッド数の監視を行いません。

総スレッド数 (1～100)

起動しているスレッドの総数の異常を検出するしきい値(システム上限値に対する割合)を設定します。

継続時間 (1～1440)

総スレッド数の異常を検出する時間を設定します。
指定した時間以上連続してしきい値を超過した場合、異常を検出します。

ユーザごとの起動プロセス数の監視

ユーザごとの起動プロセス数の監視を行うかどうかを設定します。

- チェックボックスがオン

ユーザごとの起動プロセス数の監視を行います。

- チェックボックスがオフ

ユーザごとの起動プロセス数の監視を行いません。

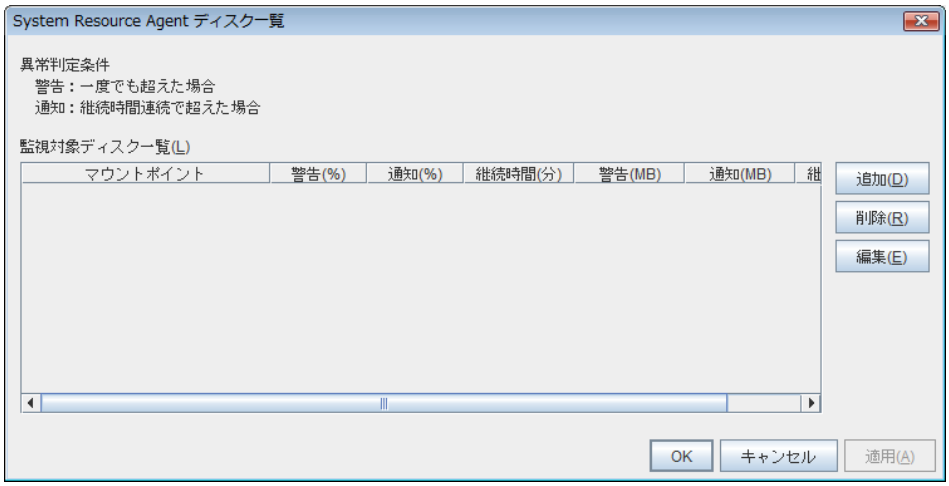
ユーザごとの起動プロセス数 (1～100)

ユーザごとの起動プロセス数の異常を検出するしきい値(システム上限値に対する割合)を設定します。

継続時間 (1～1440)

ユーザごとの起動プロセス数の異常を検出する時間を設定します。
指定した時間以上連続してしきい値を超過した場合、異常を検出します。

System Resource Agent ディスクー覧



追加

監視するディスクを追加します。[監視条件の入力] ダイアログボックスが表示されます。

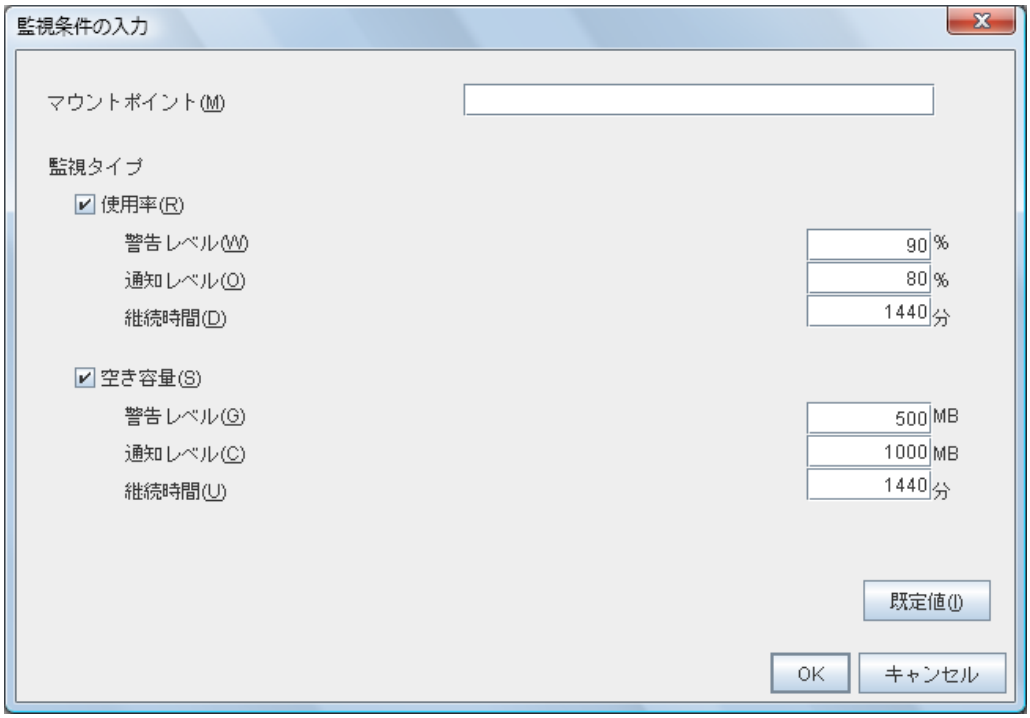
[監視条件の入力] ダイアログの説明に従い異常とする監視条件の詳細設定を行います。

削除

[ディスクー覧] で選択しているディスクを監視対象から削除します。

編集

[監視条件の入力]ダイアログボックスが表示されます。[ディスクー覧] で選択しているディスクの監視条件が表示されるので、編集して[OK]を選択します。



マウントポイント (1024 バイト以内)

監視を行うマウントポイントを設定します。[/] で始まる必要があります。

使用率

ディスク使用率の監視を行うかどうかを設定します。

- チェックボックスがオン

ディスク使用率の監視を行います。

- チェックボックスがオフ

ディスク使用率の監視を行いません。

警告レベル (1～100)

ディスク使用率の警報レベルの異常を検出するしきい値を設定します。

通知レベル (1～100)

ディスク使用率の通知レベルの異常を検出するしきい値を設定します。

継続時間 (1～43200)

ディスク使用率の通知レベルの異常を検出する時間を設定します。

指定した時間以上連続してしきい値を超過した場合、異常を検出します。

空き容量

ディスク空き容量の監視を行うかどうかを設定します。

- チェックボックスがオン

ディスク空き容量の監視を行います。

- チェックボックスがオフ

ディスク空き容量の監視を行いません。

警告レベル (1～4294967295)

ディスク空き容量の警報レベルの異常を検出する容量(MB)を設定します。

通知レベル (1～4294967295)

ディスク空き容量の通知レベルの異常を検出する容量(MB)を設定します。

継続時間 (1～43200)

ディスク空き容量の通知レベルの異常を検出する時間を設定します。

指定した時間以上連続してしきい値を超過した場合、異常を検出します。

システムモニタリソースの注意事項

システムモニタリソースを利用する場合、zip および unzip のパッケージが必要です。

監視動作ごとに System Resource Agent 自体が運用ログなどを出力することがあります。

回復対象には System Resource Agent がリソース監視の異常を検出した際のフェイルオーバ対象リソースを指定してください。

System Resource Agent の設定値は、デフォルトで使用することを推奨します。

以下のような場合には、リソース監視の異常を検出できないことがあります。

・ システム全体のリソース監視で、しきい値をはさんで増減を繰り返している場合

スワップアウトされているプロセスについては、リソース異常の検出対象になりません。

動作中に OS の日付/時刻を変更した場合、10 分間隔で行っている解析処理のタイミングが日付/時刻変更後の最初の一回だけずれてしまいます。以下のようなことが発生するため、必要に応じてクラスタのサスペンド・リジュームを行ってください。

・ 異常として検出する経過時間を過ぎても、異常検出が行われない。

・ 異常として検出する経過時間前に、異常検出が行われる。

クラスタのサスペンド・リジュームを行った場合、その時点から情報の収集を開始します。

SELinux の設定は permissive または disabled にしてください。

enforcing に設定すると CLUSTERPRO で必要な通信が行えない場合があります。

プロセスリソース、システムリソースの使用量の解析は 10 分間隔で行います。そのため、監視継続時間を経過してから最大 10 分後に異常を検出する場合があります。

ディスクリソースの使用量の解析は 60 分間隔で行います。そのため、監視継続時間を経過してから最大 60 分後に異常を検出する場合があります。

ディスクリソースの空き容量監視にて指定するディスクサイズは、実際のディスクサイズより小さい値を指定してください。大きい値を指定した場合、空き容量不足として異常検出します。

監視中のディスクを交換した場合、交換前と交換後のディスクにて以下のいずれかが異なる場合、それまでの解析情報はクリアします。

・ ディスクの総容量

・ ファイルシステム

スワップ領域を割り当てていないマシンでは、システムの総仮想メモリ使用量の監視のチェックを外してください。

ディスクリソース監視機能は、ディスクデバイス以外は監視対象外です。

System Resource Agent で収集しているディスク使用率は、ディスク総容量とディスク使用可能容量で計算しています。df(1) コマンドで表示されるディスク使用率とは算出方法の違いにより、値が若干異なる場合があります。

ディスクリソース監視機能で同時に監視できる最大のディスク数は 64 台です。

システムモニタリソースは収集した統計情報および解析情報をファイル出力します。これらのファイル数が下記最大個数に達した場合には、古いファイルから削除を行います。
(下記文中の <data パス> は <インストールディレクトリ>/ha/sra/data/ となります。)

・システムリソース、プロセスリソースの統計情報

パス: <data パス>/hasrm_monitor_list.xml.YYYYMMDDhhmmss.zip

最大個数: 1500 個

・システムリソース、プロセスリソースの解析情報

パス: <data パス>/hasrm_analyze_list.xml.YYYYMMDDhhmmss.zip

最大個数: 3 個

・ディスクリソースの統計情報

パス: <data パス>/hasrm_diskcapacity_monitor_list.xml.YYYYMMDDhhmmss.zip

最大個数: 10 個

・ディスクリソースの解析情報

パス: <data パス>/hasrm_diskcapacity_analyze_list.xml.YYYYMMDDhhmmss.zip

最大個数: 3 個

システムモニタリソースの監視方法

システムモニタリソースは、以下の監視を行います。

プロセス、システムおよびディスクのリソースの使用量を継続的に収集し、解析します。

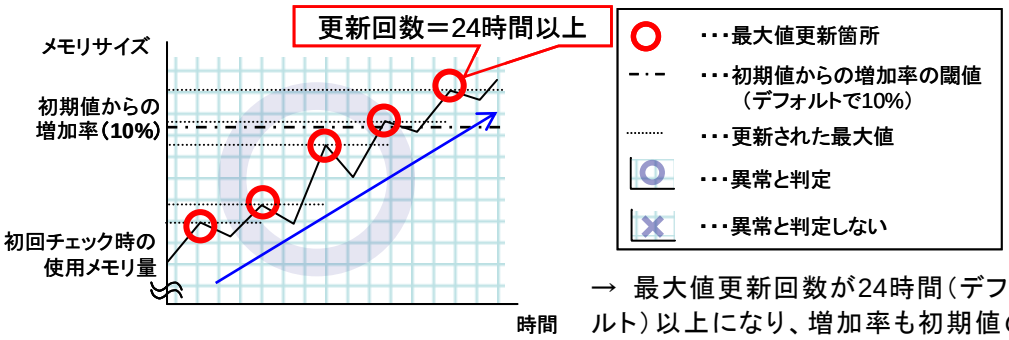
リソースの使用量があらかじめ設定したしきい値以上になった場合、異常を検出します。

異常を検出した状態が監視継続時間連続すると、リソース監視の異常を通知します。

プロセスリソース監視（CPU、メモリ、オープンファイル数、ゾンビプロセス）をデフォルト値で運用した場合、24 時間後にリソース監視の異常を通知します。

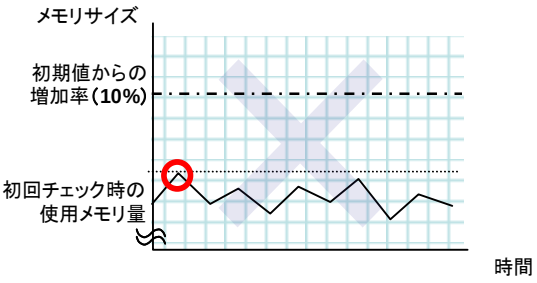
以下に、プロセスリソース監視のメモリ使用量の異常検出の例を示します。

◆ メモリ使用量が経過時間と共に増減しながら、規定回数以上最大値を更新し、増加率が初期値の 10% 以上になった



→ 最大値更新回数が24時間（デフォルト）以上になり、増加率も初期値の10%を上回っているため、メモリリークと判定します。

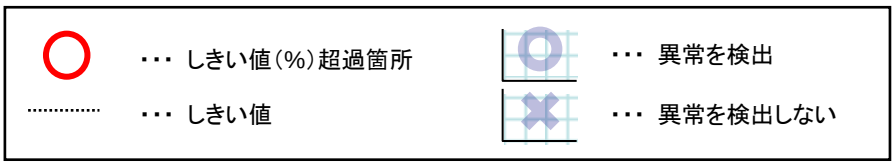
◆ メモリ使用量が経過時間と共に一定の範囲内で増減



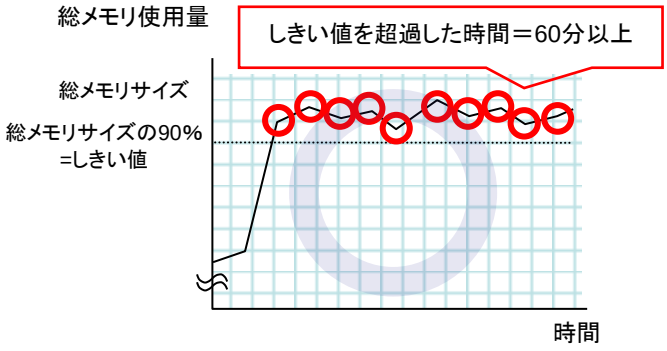
→ メモリ使用量は、一定の値未満の範囲で増減しているため、メモリリークと判定しません。

システムリソース監視をデフォルト値で運用した場合、リソースの使用量が 90% 以上の状態が連続すると、 60 分後にリソース監視の異常を通知します。

以下に、システムリソース監視をデフォルト値で運用した場合の総メモリ使用量の異常検出の例を示します。

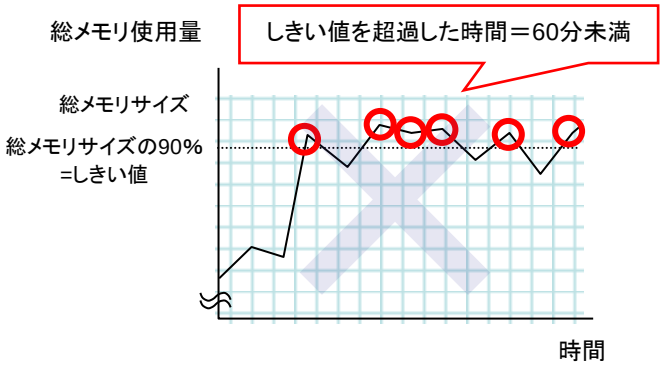


◆ 総メモリ使用量が経過時間と共に総メモリ使用量のしきい値以上の状態が続き、一定時間以上になった



→ 総メモリ使用量がしきい値(90%)以上の状態が続き、連続して監視継続時間(60分)以上になったため、総メモリ使用量の異常を検出します。

◆ 総メモリ使用量が経過時間と共に総メモリ使用のしきい値の前後で増減し、連続して総メモリ使用量のしきい値以上にならない



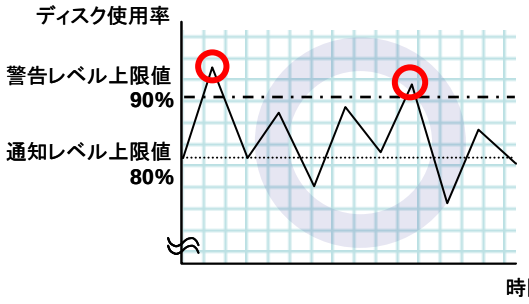
→ 総メモリ使用量は一時的に総メモリ使用量のしきい値(90%)以上になるが、監視継続時間(60分)連続してしきい値以上になる前にしきい値未満で推移しているため、総メモリ使用量の異常を検出しません。

ディスクリソース監視をデフォルト値で運用した場合、 24 時間後に通知レベルの異常を通知します。

以下に、ディスクリソース監視をデフォルト値で運用した場合のディスク使用率の異常検出の例を示します。

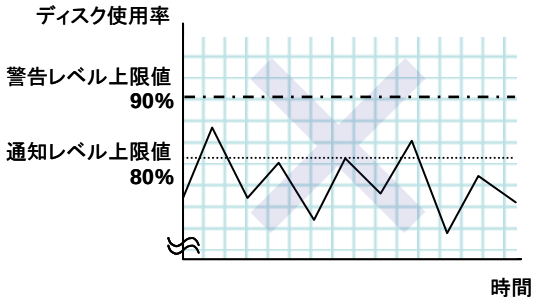
警告レベルのディスク容量監視

◆ ディスク使用率が警告レベル上限値で指定された一定のしきい値以上になった



→ ディスク使用率が警告レベル上限値を超えたため、ディスク容量監視異常と判定します。

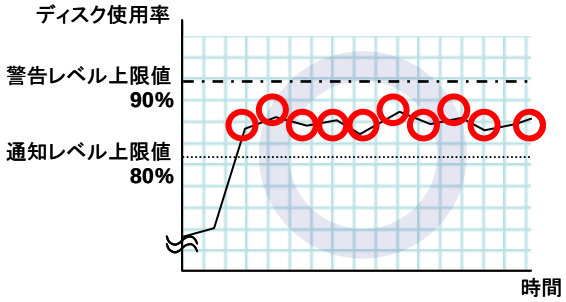
◆ ディスク使用率が一定の範囲内で増減し、警告レベル上限値で指定された一定のしきい値以上にならない



→ ディスク使用率は警告レベル上限値を超えない範囲で増減しているため、ディスク容量監視異常と判定しません。

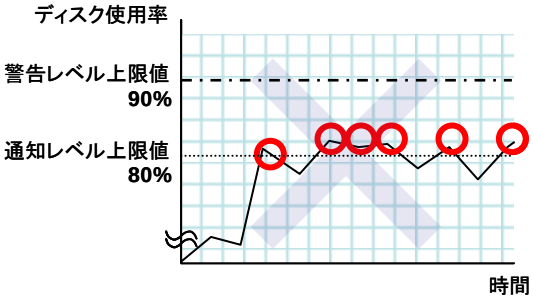
通知レベルのディスク容量監視

◆ ディスク使用率が経過時間と共に通知レベル上限値で指定された一定のしきい値以上の状態が続き、一定時間以上になった



→ ディスク使用率が通知レベル上限値を連続して超えたため、ディスク容量監視異常と判定します。

◆ ディスク使用率が一定の範囲内で増減し、通知レベル上限値で指定された一定のしきい値以上にならない

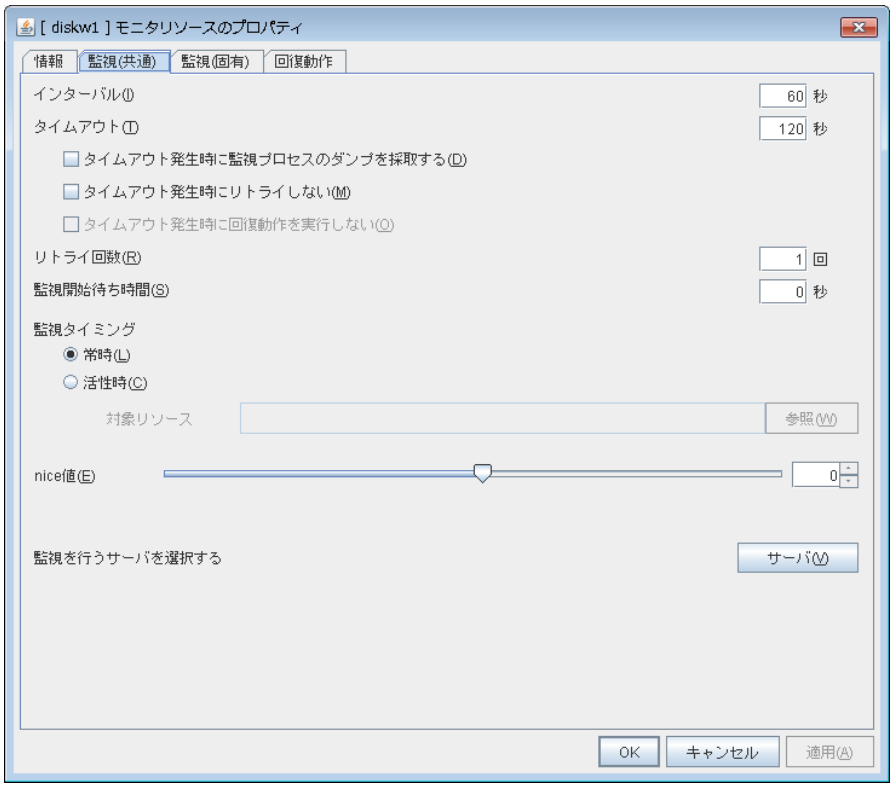


→ ディスク使用率は通知レベル上限値の前後で増減しているため、ディスク容量監視異常と判定しません。

モニタリソース共通の設定

この設定の項目はモニタリソース共通です。

1. 監視処理の設定



インターバル(1～999)

監視対象の状態を確認する間隔を設定します。

タイムアウト(5～999 [3])

ここで指定した時間内に監視対象の正常状態が検出できない場合に異常と判断します。

タイムアウト発生時に監視プロセスのダンプを採取する

本機能を有効にした場合、モニタリソースがタイムアウトすると、タイムアウトしたモニタリソースのダンプが採取されます。採取されたダンプ情報は、/opt/nec/clusterpro/work/rm/”モニタリソース名”/errinfo.cur フォルダ配下に保存されます。採取が複数回実行された場合は、過去の採取情報のフォルダ名が errinfo.1, errinfo.2 とリネームされます。ダンプ情報は最大 5 回採取されます。

[3] ユーザ空間モニタリソースで監視方法にipmiを設定している場合は、255以下の値を設定する必要があります。

タイムアウト発生時にリトライしない

本機能を有効にした場合、モニタリソースがタイムアウトすると即座に回復動作を実行します。

タイムアウト発生時に回復動作を実行しない

本機能を有効にした場合、モニタリソースがタイムアウトした場合に回復動作を実行しません。

[タイムアウト発生時にリトライしない] 機能を有効にしている場合のみ設定可能です。

注:

下記のモニタリソースでは、[タイムアウト発生時にリトライしない], [タイムアウト発生時に回復動作を実行しない] 機能は設定できません。

- ユーザ空間モニタリソース
- カスタムモニタリソース (監視タイプが [非同期] の場合のみ)
- マルチターゲットモニタリソース
- 仮想マシンモニタリソース
- 外部連携モニタリソース
- JVM モニタリソース
- システムモニタリソース

リトライ回数(0～999)

異常状態を検出後、連続してここで指定した回数の異常を検出したときに異常と判断します。

0 を指定すると最初の異常検出で異常と判断します。

監視開始待ち時間(0～9999)

監視を開始するまでの待ち時間を設定します。

注:

下記のモニタリソースでは、監視を開始するまでの待ち時間として、モニタタイムアウト時間 と 監視開始待ち時間の値が大きいほうを使用します。

- 外部連携モニタリソース
- カスタムモニタリソース (監視タイプが [非同期] の場合のみ)
- DB2 モニタリソース
- システムモニタリソース
- JVM モニタリソース
- MySQL モニタリソース
- Oracle モニタリソース
- PostgreSQL モニタリソース

- プロセス名モニタリソース
- Sybase モニタリソース

監視タイミング

監視のタイミングを設定します。

[常時]
監視を常時行います。

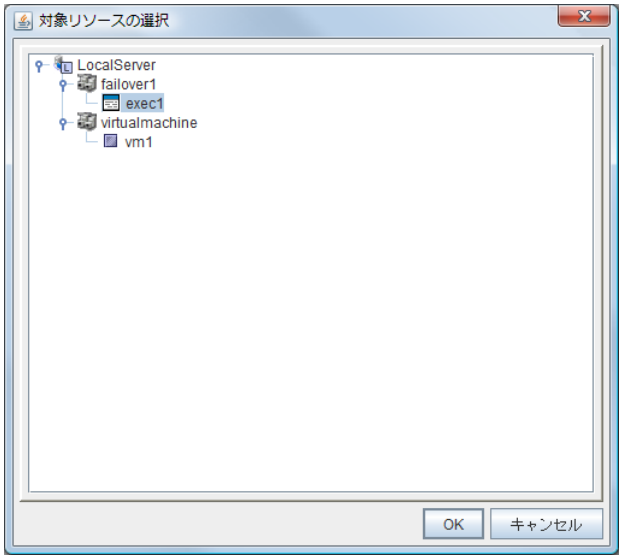
[活性時]
指定したリソースが活性するまで監視を行いません。

対象リソース

活性時監視を行う場合に対象となるリソースを表示します。

参照

対象リソースの選択ダイアログ ボックスを表示します。LocalServer とサーバに登録されているグループ名、リソース名がツリー表示されます。対象リソースとして設定するリソースを選択して[OK]をクリックします。

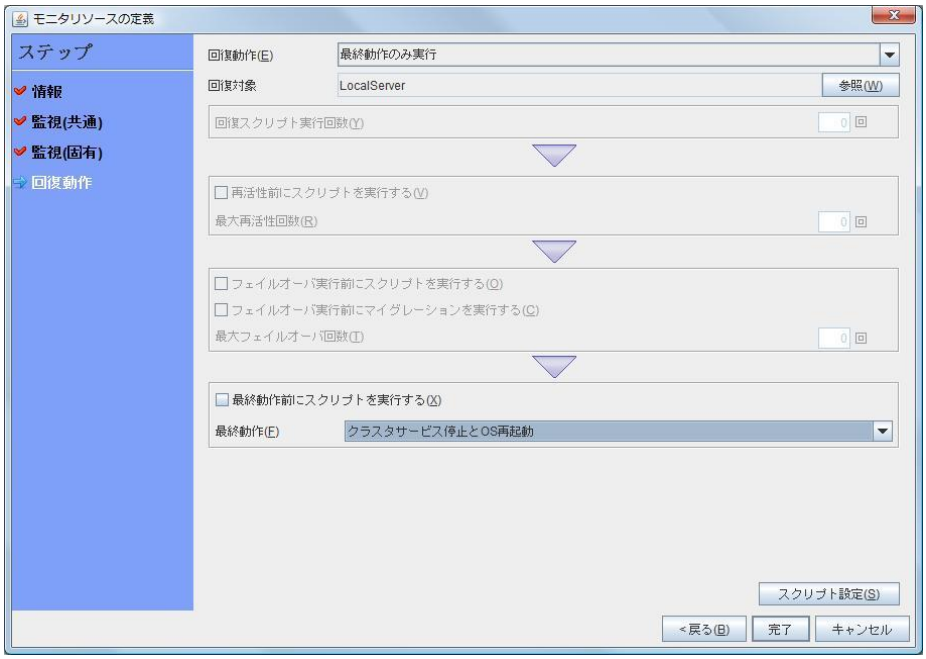


nice 値

プロセスの nice 値を設定します。

2. 復旧処理の設定

回復対象と異常検出時の動作を設定します。異常検出時にグループの再起動やリソースの再起動、サーバの再起動ができます。ただし、回復対象が非活性状態であれば回復動作は行われません。



回復動作

異常検出時の回復動作を選択します。

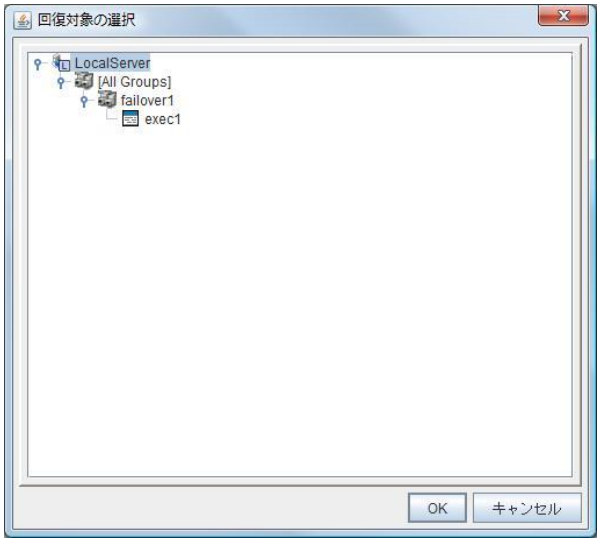
- [回復対象を再起動]
  回復対象として選択されたグループまたはグループリソースを再活性します。再活性が失敗するか、再活性後に同じ異常が検出された場合は、最終動作として選択された動作を実行します。
- [最終動作のみ実行]
  最終動作として選択された動作を実行します。
- [カスタム設定]
  回復対象として選択されたグループまたはグループリソースを最大再活性回数まで再活性します。再活性が失敗するか、再活性後に同じ異常が検出される状態が継続し、最大再活性回数に達した場合は、最終動作として選択された動作を実行します。

回復対象

リソースの異常とみなした時に回復を行う対象のオブジェクトが表示されます。

参照

回復対象の選択ダイアログ ボックスを表示します。LocalServer、All Groups とサーバに登録されているグループ名、リソース名がツリー表示されます。回復対象として設定するものを選択して[OK]をクリックします。



回復スクリプト実行回数 (0～99)

異常検出時に [スクリプト設定] で設定されたスクリプトを実行する回数を設定します。0 を設定するとスクリプトを実行しません。

再活性前にスクリプトを実行する

- チェックボックスがオン
  再活性化を実施する前にスクリプト/コマンドを実行します。スクリプト/コマンドの設定を行うためには [スクリプト設定] をクリックしてください。
- チェックボックスがオフ
  スクリプト/コマンドを実行しません。

最大再活性回数(0～99)

異常検出時に再活性化を行う回数を設定します。0 を設定すると再活性化を行いません。回復対象にグループまたはグループリソースを選択した場合に設定可能です。

フェイルオーバ実行前にスクリプトを実行する

使用しません。

フェイルオーバ実行前にマイグレーションを実行する

使用しません。

最大フェイルオーバ回数

使用しません。

最終動作前にスクリプトを実行する

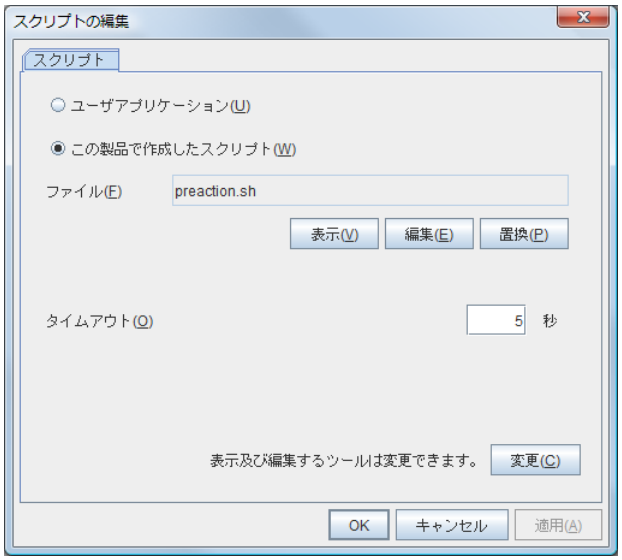
最終動作を実行する前にスクリプトを実行するかどうかを指定します。

- チェックボックスがオン
  最終動作を実施する前にスクリプト/コマンドを実行します。スクリプト/コマンドの設定を行うためには[スクリプト設定]ボタンをクリックしてください。
- チェックボックスがオフ
  スクリプト/コマンドを実行しません。

[スクリプト設定] をクリックすると、[スクリプトの編集] ダイアログボックスが表示されます。実行するスクリプトまたは実行ファイルを設定して [OK] をクリックします。

スクリプト設定

[スクリプトの編集] ダイアログボックスを表示します。回復スクリプト、回復動作前に実行するスクリプト/コマンドを設定します。



ユーザアプリケーション

スクリプトとしてサーバ上の実行可能ファイル(実行可能なシェルスクリプトファイルや実行ファイル)を使用します。ファイル名にはサーバ上のローカルディスクの絶対パスまたは実行可能ファイル名を設定します。また、絶対パスやファイル名に空欄が含まれる場合は、下記のように、ダブルクォーテーション (“) でそれらを囲ってください。

例:

“/tmp/user application/script.sh”

各実行可能ファイルは、Builder の構成情報には含まれません。Builder で編集やアップロードはできませんので、サーバ上に準備する必要があります。

この製品で作成したスクリプト

スクリプトとして Builder で準備したスクリプトファイルを使用します。必要に応じて Builder でスクリプトファイルを編集できます。スクリプトファイルは、構成情報に含まれます。

ファイル(1023 バイト以内)

[ユーザアプリケーション]を選択した場合に、実行するスクリプト（実行可能なシェルスクリプトファイルや実行ファイル）を設定します。

表示

[この製品で作成したスクリプト]を選択した場合に、スクリプトファイルをエディタで表示します。エディタで編集して保存した内容は反映されません。表示しようとしているスクリプトファイルが表示中または編集中の場合は表示できません。

編集

[この製品で作成したスクリプト]を選択した場合に、スクリプトファイルをエディタで編集します。変更を反映するには上書き保存を実行してください。編集しようとしているスクリプトファイルが既に表示中または編集中の場合は編集できません。スクリプトファイル名の変更はできません。

置換

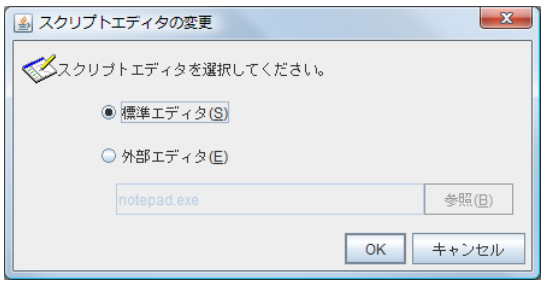
[この製品で作成したスクリプト]を選択した場合に、スクリプトファイルの内容を、ファイル選択ダイアログ ボックスで選択したスクリプトファイルの内容に置換します。スクリプトが既に表示中または編集中の場合は置換できません。ここではスクリプトファイルを選択してください。バイナリファイル(アプリケーションなど)は選択しないでください。

タイムアウト(1～9999)

スクリプトの実行完了を待ち合わせる最大時間を指定します。既定値は 5 秒です。

変更

スクリプトエディタの変更ダイアログが表示されます。スクリプトを表示または編集するエディタを任意のエディタに変更できます。



標準エディタ

スクリプトエディタに標準のエディタを使用します。

- Linux … vi(実行ユーザのサーチパスで検索される vi)
- Windows … メモ帳(実行ユーザのサーチパスで検索される notepad.exe)

外部エディタ

スクリプトエディタを任意に指定します。[参照]を選択し、使用するエディタを指定します。

Linux でCUIベースのエディタを外部エディタで指定するにはシェルスクリプトを作成してください。

以下の例は viを実行するシェルスクリプトです。

xterm -name clpedit -title "Cluster Builder" -n "Cluster Builder" -e vi "$1"

最終動作

再活性化による回復が失敗した後の回復動作を選択します。

最終動作は以下の動作が選択できます。

- 何もしない

  何も行いません。

  注:

  [何もしない] の設定は

  - 一時的に最終動作を抑止したい場合
  - 異常を検出したときにアラートの表示のみを行いたい場合
  - 実際の最終動作はマルチターゲットモニタリソースで行いたい場合

  に使用してください。

- リソース停止

  回復対象としてグループリソースが選択されている場合、選択したグループリソースとそのグループリソースに依存するグループリソースを停止します。
  回復対象に ”LocalServer” 、”All Groups”、グループが選択されている場合は選択できません。

- グループ停止

  監視対象としてグループが選択されている場合そのグループを、また監視対象としてグループリソースが選択されている場合そのグループリソースが所属するグループを停止します。”All Groups”が選択されている場合は、モニタリソースが異常を検出したサーバで起動している全てのグループを停止します。

- クラスタサービス停止

  CLUSTERPRO X SingleServerSafe を停止します。

- クラスタサービス停止とOSシャットダウン

  CLUSTERPRO X SingleServerSafe を停止し、OS をシャットダウンします。

- クラスタサービス停止とOS再起動

  CLUSTERPRO X SingleServerSafe を停止し、OS を再起動します。

- sysrq パニック

  sysrq のパニックを行います。

  注: sysrq パニックに失敗した場合、OS のシャットダウンを行います。

- keepalive リセット

  clpkhb ドライバ、clpkaドライバを使用し、OS をリセットします。

注: keepalive リセットに失敗した場合、OS のシャットダウンを行います。
clpkhb ドライバ、clpka ドライバが対応していない OS、kernel では設定しないでください。

- keepalive パニック

  clpkhb ドライバ、clpka ドライバを使用し、OS をパニックします。

  注: keepalive パニックに失敗した場合、OS のシャットダウンを行います。
  clpkhb ドライバ、clpka ドライバが対応していない OS、kernel では設定しないでください。

- BMC リセット

  ipmi のコマンドを使用し、サーバをハードウェアリセットします。

  注: BMC リセットに失敗した場合、OS のシャットダウンを行います。

  ipmitool または ipmiutil をインストールしていない、または ipmitool コマンド、hwreset コマンド、ireset コマンドが動作しないサーバでは設定しないでください。

- BMC パワーオフ

  ipmi のコマンドを使用し、OS の電源をオフにします。OS の ACPI の設定により OS のシャットダウンが実行される場合があります。

  注: BMC パワーオフに失敗した場合、OS のシャットダウンを行います。

  ipmitool または ipmiutil をインストールしていない、または ipmitool コマンド、hwreset コマンド、ireset コマンドが動作しないサーバでは設定しないでください。

- BMC パワーサイクル

  ipmi のコマンドを使用し、サーバのパワーサイクル(電源オフ/オン)を実行します。OS の ACPI の設定により OS のシャットダウンが実行される場合があります。

  注: BMC パワーサイクルに失敗した場合、OS のシャットダウンを行います。

  ipmitool または ipmiutil をインストールしていない、または ipmitool コマンド、hwreset コマンド、ireset コマンドが動作しないサーバでは設定しないでください。

- BMC NMI

  ipmi のコマンドを使用し、サーバへ NMI を発生させます。NMI 発生後の挙動は OS の設定に依存します。

  注: BMC NMI に失敗した場合、OS のシャットダウンを行います。
  ipmitool または ipmiutil をインストールしていない、または ipmitool コマンド、hwreset コマンド、ireset コマンドが動作しないサーバでは設定しないでください。

第 6 章　　ハートビートリソースの詳細

本章では、ハートビートリソースの詳細について説明します。

ハートビートリソース一覧

サーバの死活監視を行います。ハートビートデバイスには以下の種類があります。

ハートビートリソース名	略称	機能概要
LANハートビートリソース	lanhb	LANを使用してサーバの死活監視を行います

- LANハートビートは1つ設定する必要があります。

LAN ハートビートリソースの設定

LANハートビートリソースの注意事項

- LANハートビートリソースは1つ設定する必要があります。

第 7 章　その他の設定の詳細

本章では、CLUSTERPRO X SingleServerSafe のその他の項目についての詳細を説明します。

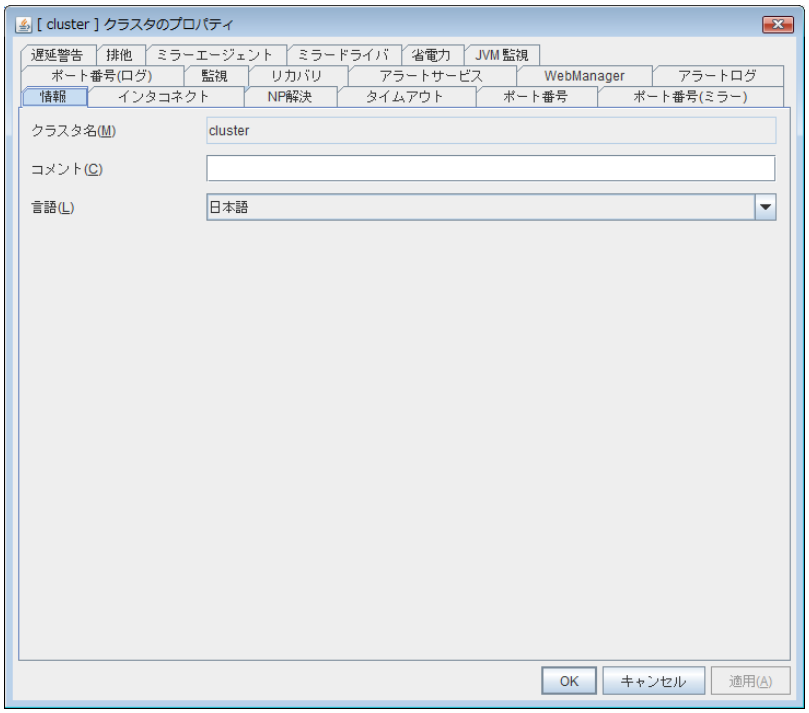
本章で説明する項目は以下のとおりです。

クラスタプロパティ

「クラスタのプロパティ」では、CLUSTERPRO X SingleServerSafe の詳細情報の表示や設定変更ができます。

情報タブ

サーバ名の表示、コメントの登録、変更を行います。



名前

サーバ名を表示します。ここでは名前の変更はできません。

コメント(127 バイト以内)

コメントを設定します。半角英数字のみ入力可能です。

言語

表示言語を以下の中から選択します。WebManager を動作させる OS の言語(ロケール)に設定してください。

- 英語
- 日本語
- 中国語

インタコネクトタブ

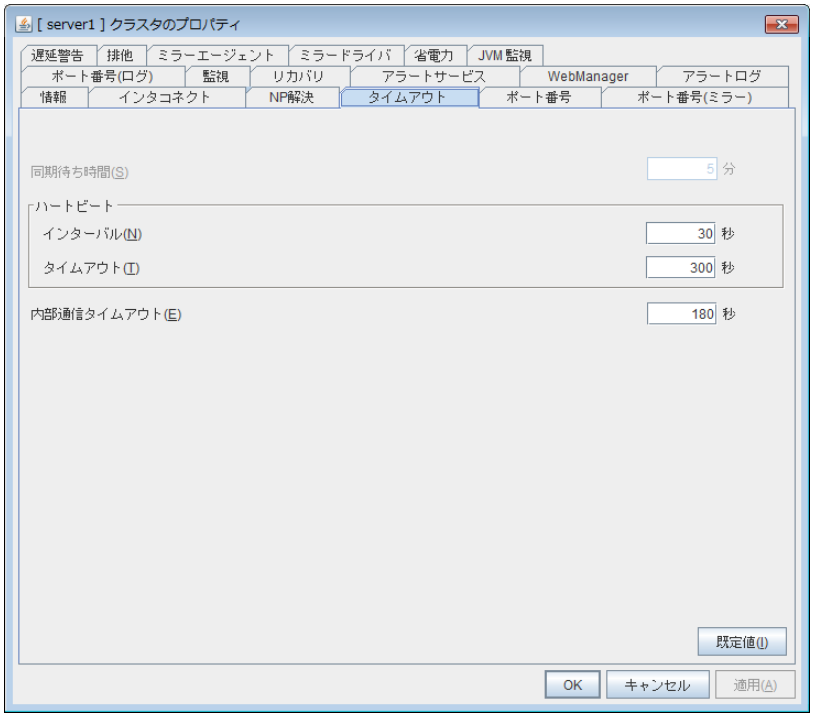
使用しません。

NP 解決タブ

使用しません。

タイムアウトタブ

タイムアウトなどの値を設定します。



同期待ち時間 (0～99)

使用しません。

ハートビート

ハートビート間隔および、ハートビートタイムアウトです。

- インターバル (1～99)

  ハートビートの間隔です。

- タイムアウト (2～9999)

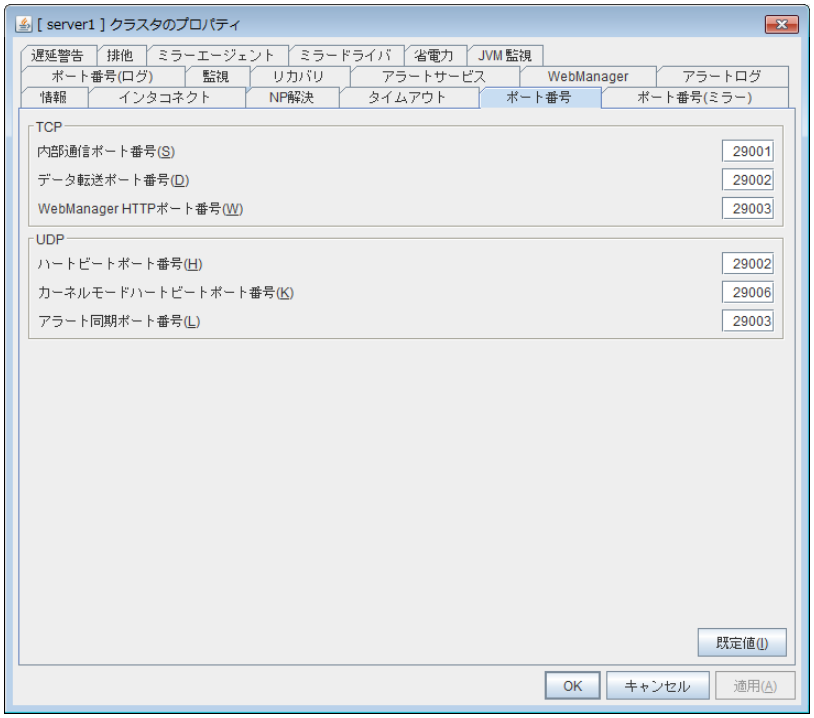
  ハートビートタイムアウトです。ここで設定された時間の間無応答が続くとサーバダウンとみなします。

  - インターバルより大きい値である必要があります。

- シャットダウン監視 (284ページの「監視タブ」参照) をする場合は、アプリケーションを含めて OS がシャットダウンする時間より長い時間にする必要があります。

ポート番号タブ

TCP ポート番号、UDP ポート番号を設定します。



TCP

TCP の各ポート番号は重複できません。

- 内部通信ポート番号(1～65535 [4])
  内部通信で使うポート番号です。
- データ転送ポート番号(1～65535 [4])
  トランザクション(構成情報反映/バックアップ、ライセンス情報送受信、コマンド実行)で使うポート番号です。
- WebManager HTTP ポート番号(1～65535 [4])
  ブラウザが CLUSTERPRO サーバと通信するときに使うポート番号です。

UDP

UDP の各ポート番号は重複できません。

- カーネルモードハートビートポート番号(1～65535 [4])
  カーネルモードハートビートで使うポート番号です。
  使用しません。
- アラート同期ポート番号(1～65535 [4])
  サーバ間でアラートメッセージを同期するときに使うポート番号です。

既定値

[4] Well-knownポート、特に 1～1023番の予約ポートの使用は推奨しません。

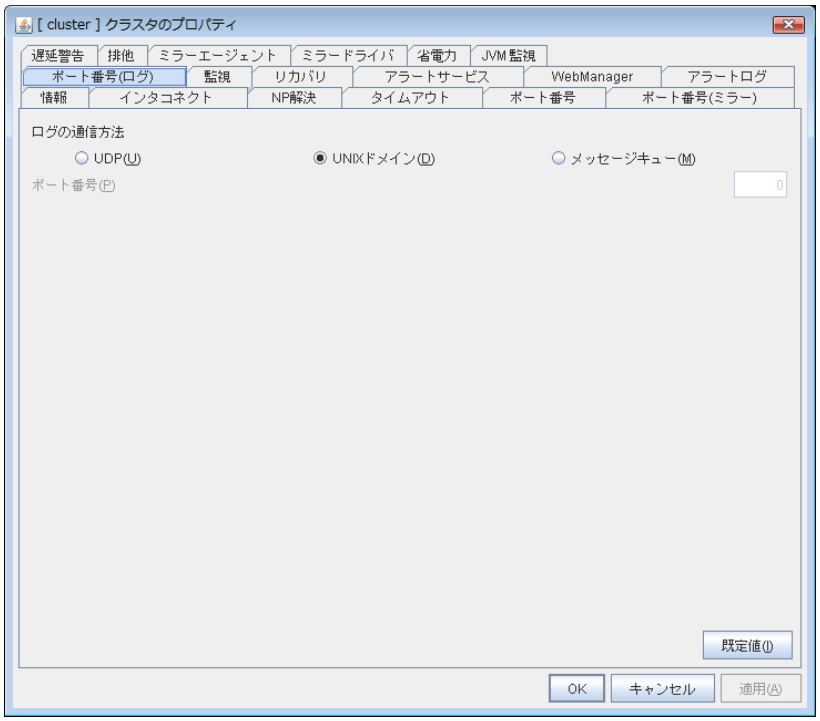
既定値に戻すときに使用します。[既定値]ボタンを選択すると全ての項目に既定値が設定されます。

ポート番号(ミラー) タブ

使用しません。

ポート番号(ログ) タブ

ログの通信方法を設定します。



ログの通信方法

- UDP
  ログの通信方法に UDP を使用します。
- UNIX ドメイン
  ログの通信方法に UNIX ドメインを使用します。
- メッセージキュー
  ログの通信方法に メッセージキューを使用します。

注: UDP は、SuSE Linux Enterprise Server 11 では使用できません。

ポート番号 (1～65535)

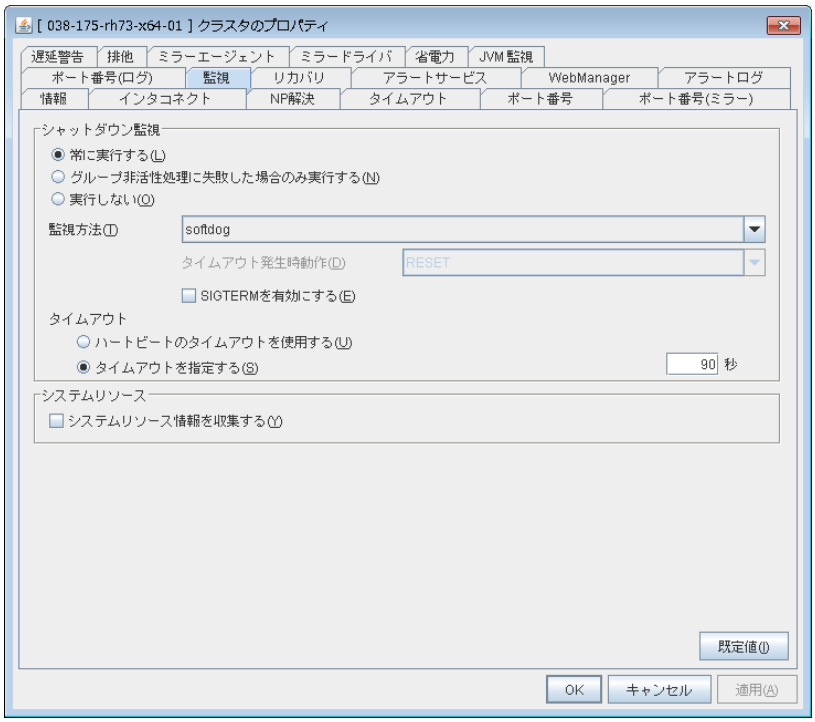
ログの通信方法で UDP を選択した場合に使うポート番号です。[ポート番号] タブの UDP の各ポート番号と重複することはできません。

既定値

既定値に戻すときに使用します。[既定値] をクリックすると全ての項目に既定値が設定されます。

監視タブ

監視に関する設定をします。



シャットダウン監視

CLUSTERPRO のコマンドでサーバシャットダウンを実行したときに、OS がストールしているか否か監視します。
クラスタサービスは OS がストールしていると判断すると強制的にサーバをリセットまたはパニックします。サーバのパニックは、監視方法 keepalive の場合のみ設定可能です。

- 常に実行する

  シャットダウン監視をします。ハートビートタイムアウト (279ページの「タイムアウトタブ」参照) を、アプリケーションを含めて OS がシャットダウンする時間より長い時間にする必要があります。

- グループ非活性処理に失敗した場合のみ実行する

  グループの非活性に失敗した場合のみシャットダウン監視をします。ハートビートタイムアウト (279ページの「タイムアウトタブ」参照) を、アプリケーションを含めて OS がシャットダウンする時間より長い時間にする必要があります。

- 実行しない

  シャットダウン監視をしません。

監視方法

シャットダウン監視を行う場合の監視方法を以下の中から選択します。

- softdog
- ipmi
- keepalive

タイムアウト発生時動作

OS がストールしていると判断した場合の動作を以下の中から選択します。監視方法に keepalive を選択した場合のみ設定できます。

- RESET
  サーバをリセットします
- PANIC
  サーバをパニックさせます

SIGTERM を有効にする

シャットダウン監視を行う場合に SIGTERM を有効にするかどうかを設定します。

注: [監視方法] で ipmi を選択して、[SIGTERM を有効にする] をオフに設定にしている場合、OS のシャットダウンが正常に終了してもリセットすることがあります。

ハートビートタイムアウトを使用する

シャットダウン監視のタイムアウト値をハートビートタイムアウト値と連動させます。

タイムアウト (2～9999)

シャットダウン監視のタイムアウト値としてハートビートタイムアウト値を使用しない場合にタイムアウト値を指定します。

システムリソース

システムリソース情報を収集する/しないを設定します。

運用性向上のためにシステムリソース情報を定期的に収集します。システムリソース情報は、CLUSTERPRO の動作状況の調査に役立ち、システムリソース不足を起因とする障害の原因特定が容易になります。

チェックボックスがオン

サーバ動作中に CPU やメモリ、プロセスなどのシステムリソース情報を定期的に収集します。

収集したシステムリソース情報は clplogcc コマンドや Webmanager によるログ収集で収集されます。

clplogcc コマンドでのログ収集時には type2 を、WebManager でのログ収集時にはパターン 2 を指定してください。

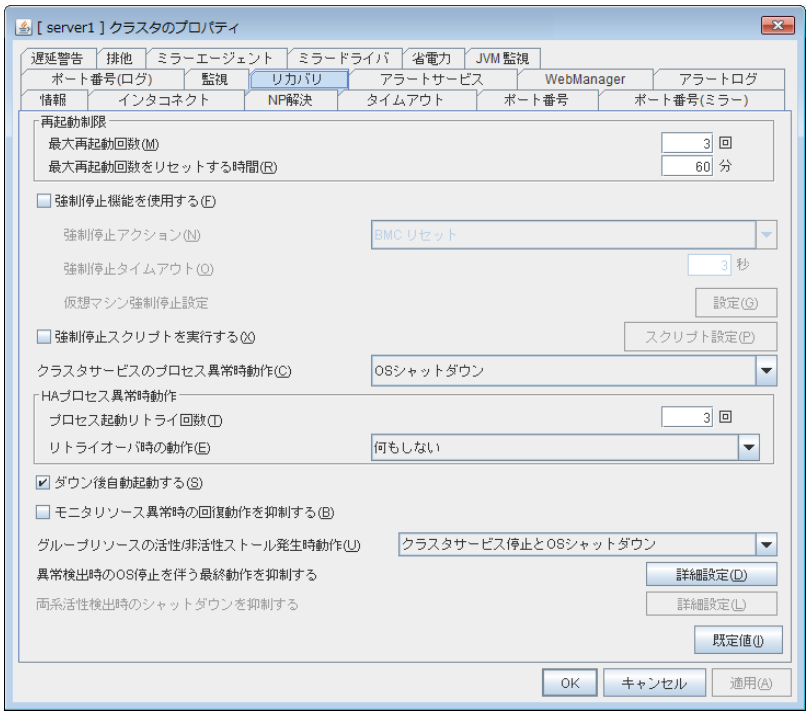
プロセスの起動数などのシステム稼働状況に依存しますが、リソース情報の保存には 450MB 以上のディスク領域が必要となります。

チェックボックスがオフ

システムリソース情報を収集しません。

リカバリタブ

リカバリに関する設定をします。



再起動制限

グループリソースとモニタリソースの異常検出時の最終動作として、OS の再起動を伴うような設定をしている場合、永遠に再起動を繰り返してしまうことがあります。再起動の回数を設定することによって再起動の繰り返しを制限できます。

- 最大再起動回数(0～99)

  再起動の制限回数を設定します。ここで指定する回数はグループリソース、モニタリソースで別々にカウントされます。

- 最大再起動回数をリセットする時間(0～999)

  最大再起動回数を指定している場合に、クラスタ起動時からの正常動作がここで指定した時間続いた時、それまでの再起動回数はリセットされます。ここで指定する時間はグループリソース、モニタリソースで別々にカウントされます。

注: [最大再起動回数をリセットする時間] に 0 を設定した場合、再起動回数はリセットされません。再起動回数をリセットする場合は、clpregctrl コマンドを使用してください。

強制停止機能を使用する

使用しません。

強制停止アクション

使用しません。

強制停止タイムアウト(0～99)

使用しません。

仮想マシン強制停止設定

使用しません。

強制停止スクリプトを実行する

使用しません。

スクリプト設定

使用しません。

クラスタサービスのプロセス異常時アクション

クラスタサービスのプロセス異常時におけるアクションを指定します。

- OS シャットダウン

  OS をシャットダウンします。

- OS 再起動

  OS を再起動します。

HA プロセス異常時動作

- プロセス起動リトライ回数 (0～99)
  HA プロセス異常時の再起動回数を指定します。
- リトライオーバ時の動作
  HA プロセス異常時における動作を指定します。
  - クラスタサービス停止
    クラスタサービスを停止します。
  - クラスタサービス停止 と OS シャットダウン
    クラスタサービスを停止し、OS をシャットダウンします。
  - クラスタサービス停止と OS 再起動
    クラスタサービスを停止し、OS を再起動します。

注:HA プロセスは、システムモニタリソースや JVM モニタリソース、システムリソース情報収集機能で使用されるプロセスです。

ダウン後自動起動する

クラスタシャットダウンやクラスタ停止以外の方法でサーバを停止した場合やクラスタシャットダウンやクラスタ停止が正常に終了しなかった場合に、次回 OS 起動時にクラスタサービスを自動起動するかどうかを設定します。

モニタリソース異常時の回復動作を抑制する

- チェックボックスがオン
  モニタリソースの異常検出による回復動作を抑制します。
- チェックボックスがオフ
  モニタリソース異常検出による回復動作を抑制しません。

注: 本回復動作抑制機能は、モニタリソースの異常検出による回復動作を抑制するものです。グループリソースの活性異常時の復旧動作は行われます。

本機能は、ユーザ空間モニタリソースでは有効になりません。

外部連携監視リソースは異常検出時の回復動作の抑止の対象にはなりません。

グループリソースの活性/非活性ストール発生時動作

グループリソースの活性/非活性ストール発生時における動作を指定します。

- クラスタサービス停止と OS シャットダウン
  ストールが発生したサーバのクラスタサービスを停止し、OS をシャットダウンします。
- クラスタサービス停止と OS 再起動
  ストールが発生したサーバのクラスタサービスを停止し、OS を再起動します。
- sysrq パニック
  ストールが発生したサーバでsysrq のパニックを行います。
- keepalive リセット
  ストールが発生したサーバで clpkhb ドライバ、clpka ドライバを使用し、OS をリセットします。
- keepalive パニック
  ストールが発生したサーバで clpkhb ドライバ、clpka ドライバを使用し、OS をパニックします。
- BMC リセット
  ストールが発生したサーバで ipmi のコマンドを使用し、サーバをハードウェアリセットします。
- BMC パワーオフ
  ストールが発生したサーバで ipmi のコマンドを使用し、OS の電源をオフにします。OS の ACPI の設定により OSのシャットダウンが実行される場合があります。
- BMC パワーサイクル
  ストールが発生したサーバで ipmi のコマンドを使用し、サーバのパワーサイクル（電源オフ/オン）を実行します。OSの ACPI の設定により OS のシャットダウンが実行される場合があります。
- BMC NMI
  ストールが発生したサーバで ipmi のコマンドを使用し、サーバへ NMI を発生させます。NMI 発生後の挙動は OS の設定に依存します。
- 何もしない(活性/非活性異常として扱う)
  グループリソースの活性/非活性異常検出時の復旧動作を行います。

注:「何もしない(活性/非活性異常として扱う)」を指定してストールが発生した場合、グループリソースへの影響が不定となりますので、「何もしない(活性/非活性異常として扱う)」への設定変更は推奨しません。

「何もしない(活性/非活性異常として扱う)」を指定する場合は、グループリソースの活性/非活性異常検出時の復旧動作の設定を以下のようにしてください。

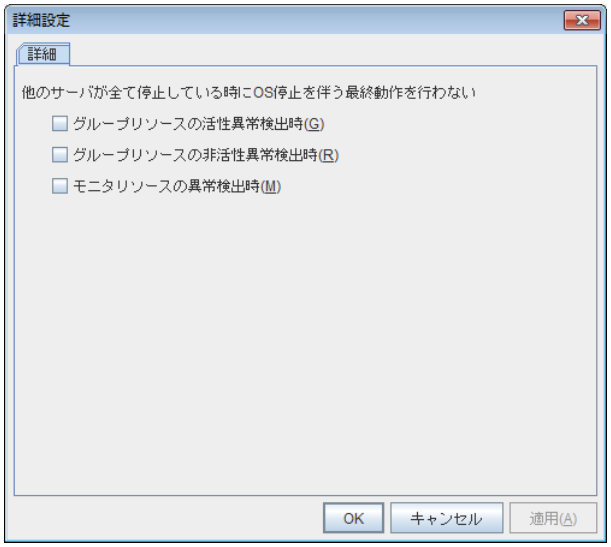
・活性/非活性リトライしきい値 ： 0 回

・フェイルオーバしきい値 ： 0 回

・最終動作 ： OS 停止を伴う動作

異常検出時の OS 停止を伴う最終動作を抑制する

[詳細設定]をクリックし、異常検出時の OS 停止を伴う最終動作の抑制を設定します。



◆ グループリソースの活性異常検出時

グループリソースの活性異常検出時の最終動作が OS 停止を伴うものに設定されている場合、活性異常検出時の最終動作が抑制されます。

◆ グループリソースの非活性異常検出時

グループリソースの非活性異常検出時の最終動作が OS 停止を伴うものに設定されている場合、非活性異常検出時の最終動作が抑制されます。

◆ モニタリソースの異常検出時

モニタリソースの異常検出時の最終動作が OS 停止を伴うものに設定されている場合、異常検出時の最終動作が抑制されます。

注:

・外部連携監視リソースは異常検出時の最終動作の抑止の対象にはなりません。

・ グループリソースの活性/非活性異常検出時の最終動作、およびモニタリソースの異常検出時の最終動作で OS 停止を伴うものは以下の通りです。

- クラスタサービス停止と OS シャットダウン
- クラスタサービス停止と OS 再起動
- sysrq パニック
- keepalive リセット
- keepalive パニック
- BMC リセット
- BMC パワーオフ
- BMC パワーサイクル
- BMC NMI

両系活性検出時のシャットダウンを抑制する。

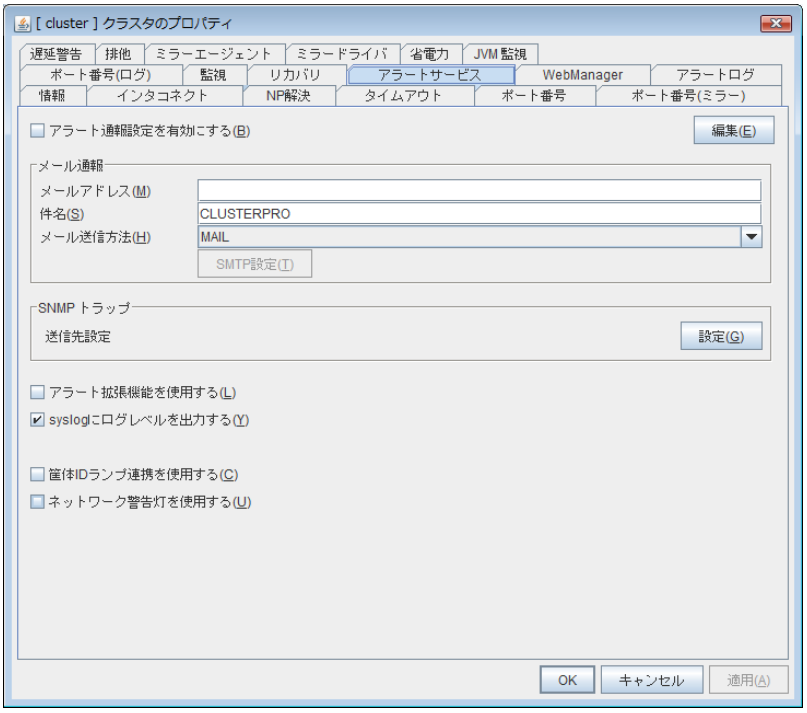
使用しません。

アラートサービスタブ

アラート通報の設定を行います。

メール通報の機能を使用する場合は、Alert Service のライセンスを登録してください。

注: メール通報機能を使用するためには CLUSTERPRO X Alert Service 3.3 for Linux を購入し、ライセンスを登録してください。



アラート通報設定を有効にする

アラート通報の設定を既定値から変更する/しない の設定をします。変更をする場合には、[編集]ボタンを押して出力先の設定をしてください。

チェックボックスをオフにすると 変更した出力先を一時的に既定値に戻すことができます。

既定の通報先は、「操作ガイド」の「syslog、アラートメッセージ一覧」を参照してください。

メールアドレス(255 バイト以内)

通報先のメールアドレスを入力します。メールアドレスを複数設定する場合は、メールアドレスをセミコロンで区切ってください。

件名(127 バイト以内)

メールの件名を入力します。

メール送信方法

メールの送信方法の設定をします。

- MAIL
  mail コマンドを使います。事前に mail コマンドでメールアドレスにメールが送信されることを確認してください。
- SMTP
  SMTP サーバと直接通信をしてメール通報します。

アラート拡張機能を使用する

CLUSTERPRO がアラートを出力する際に任意のコマンドの実行を する/しない を設定します。アラート拡張機能を使用する場合には [アラート通報設定を有効にする] のチェックボックスをオンにして[編集]ボタンを押してコマンドの設定をしてください。

チェックボックスをオフにすると 設定したコマンドを一時的に無効にすることができます。

syslog にログレベルを出力する

CLUSTERPRO X SingleServerSafe が動作中に出力する syslog のメッセージにレベルを付加します。

筐体 ID ランプ連携使用する

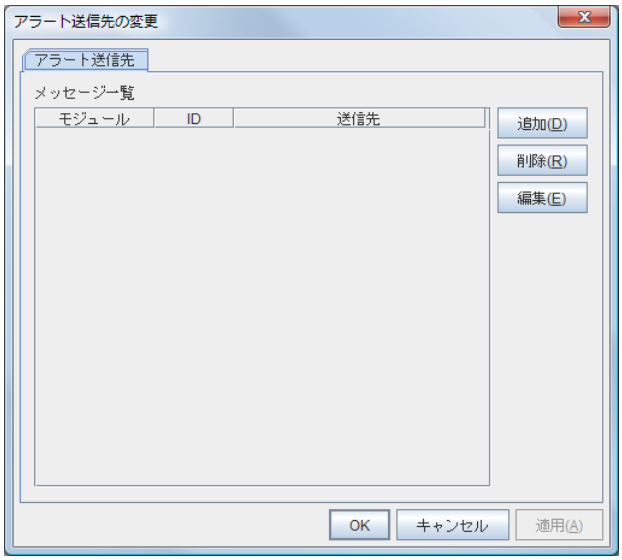
使用しません。

ネットワーク警告灯を使用する

使用しません。

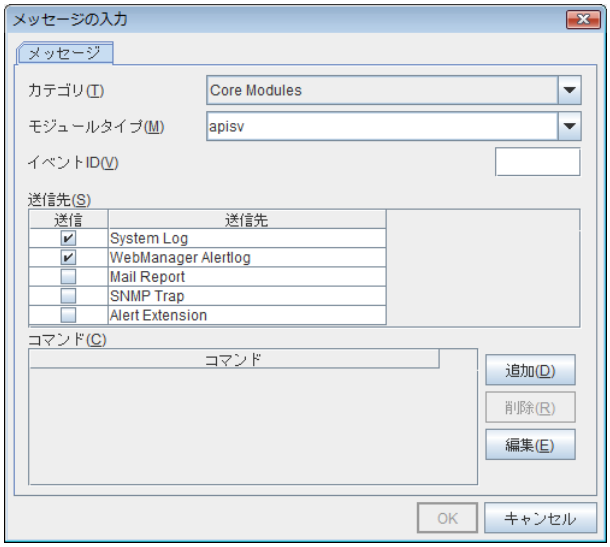
アラート送信先の変更

[編集]ボタンを選択するとアラート送信先変更ダイアログ ボックスが表示されます。



追加

送信先をカスタマイズしたいモジュールタイプ、イベント ID を追加します。[追加]ボタンを押すとメッセージの入力のダイアログが表示されます。



カテゴリ

モジュールタイプの大分類を選択します。

モジュールタイプ(31 バイト以内)

送信先を変更するモジュールタイプ名を選択します。

イベント ID

送信先を変更するモジュールタイプのイベント ID を入力します。イベント ID は「操作ガイド」の「syslog、アラートメッセージ一覧」を参照してください。

送信先

メッセージの送信として実行する処理を選択します。

- System Log
  OS の syslog へ送信します
- WebManager Alertlog
  WebManager のアラートビューにメッセージを表示します。
- Alert Extension
  指定されたコマンドを実行します（アラート拡張機能）。 [追加]ボタン、[編集]ボタンで実行するコマンドを設定・変更します。(最大 4 つのコマンドラインを指定することが出来ます)。
- Mail Report
  メール通報機能で送信します。
- SNMP Trap
  SNMP トラップ送信機能で送信します。

追加

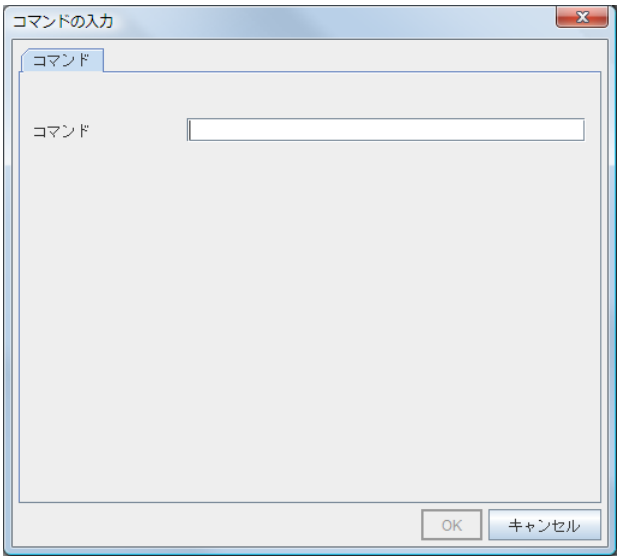
アラート拡張機能のコマンドを追加します。[追加]ボタンを押すとコマンドの入力のダイアログが表示されます。コマンドは 1 つのイベント ID について 4 個まで登録できます。

削除

アラート拡張機能のコマンドを削除する場合に使用します。コマンドを選択して、[削除]ボタンを選択してください。

編集

アラート拡張機能のコマンドを変更する場合に使用します。コマンドを選択して、[編集]ボタンを選択してください。



コマンド (511 バイト以内)

SNMP trap など通報を実行するコマンドを入力します。 絶対パスで指定してください。指定したコマンドの実行結果は参照できません。

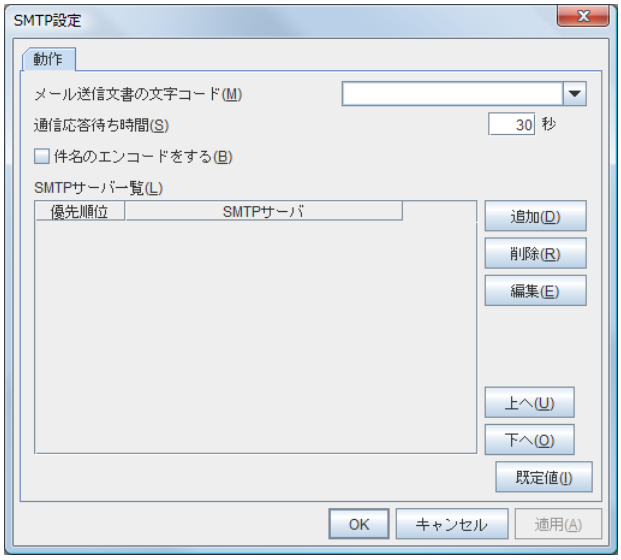
- キーワードについて
  %%MSG%% を指定すると、該当のイベント ID のメッセージ本文が挿入されます。
  1 つのコマンドに対して複数の %%MSG%% を使用することはできません。
  %%MSG%% の内容を含めて 511 バイト以内になるように設定してください。また、%%MSG%% 内に空白文字が含まれることがありますので、コマンドの引数として指定する場合には、”%%MSG%%” と指定してください。

設定例

/usr/local/bin/snmptrap -v1 -c HOME 10.0.0.2 0 10.0.0.1 1 0 ‘’ 1 s “%%MSG%%”

SMTP の設定

[SMTP 設定]ボタンを選択するとメール通報で使用する SMTP 設定ダイアログ ボックスが表示されます。



メール送信文書の文字コード(127 バイト以内)

メール通報で送信するメールの文字コードを設定します。

通信応答待ち時間(1～999)

SMTP サーバとの通信のタイムアウトを設定します。

件名のエンコードをする

メールの件名のエンコードをする/しない を設定します。

SMTP サーバの一覧

設定されている SMTP サーバを表示します。本バージョンで設定できる SMTP サーバは 1 台です。

追加

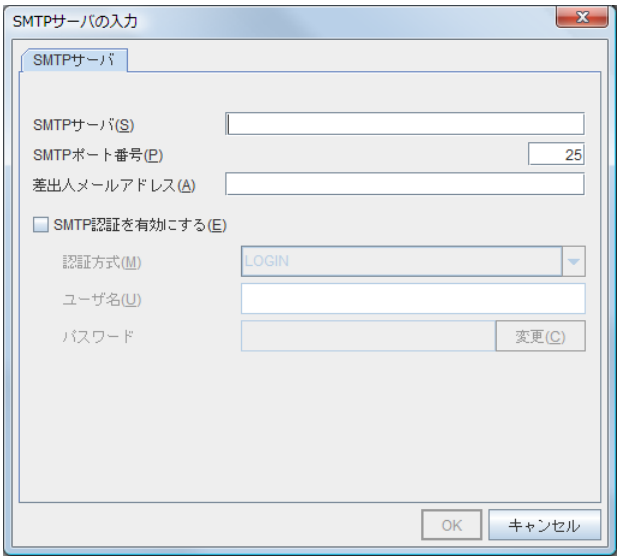
SMTP サーバを追加します。[追加]ボタンを押すと SMTP の入力のダイアログが表示されます。

削除

SMTP サーバの設定を削除する場合に使用します。

編集

SMTP サーバの設定を変更する場合に使用します。



SMTP サーバ(255 バイト以内)

SMTP サーバの IP アドレスまたはホスト名を設定します。

SMTP ポート番号(1～65535)

SMTP サーバのポート番号を設定します。

差出人メールアドレス(255 バイト以内)

メール通報で送信されるメールの送信元アドレスを設定します。

SMTP 認証を有効にする

SMTP の認証をする/しない の設定をします。

認証方式

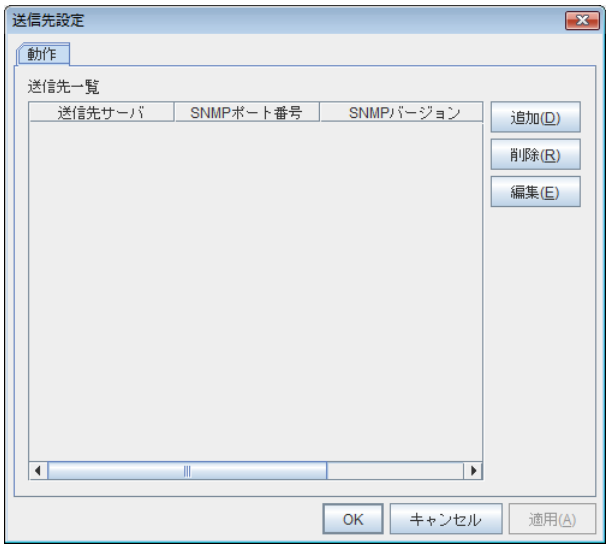
SMTP の認証の方式を選択します。

ユーザ名(255 バイト以内)

SMTP の認証で使用するユーザ名を設定します。

パスワード(255 バイト以内)

SMTP の認証で使用するパスワードを設定します。



送信先一覧

設定されている SNMP トラップ送信先を表示します。本バージョンで設定できる SNMP トラップ送信先は 32 件です。

追加

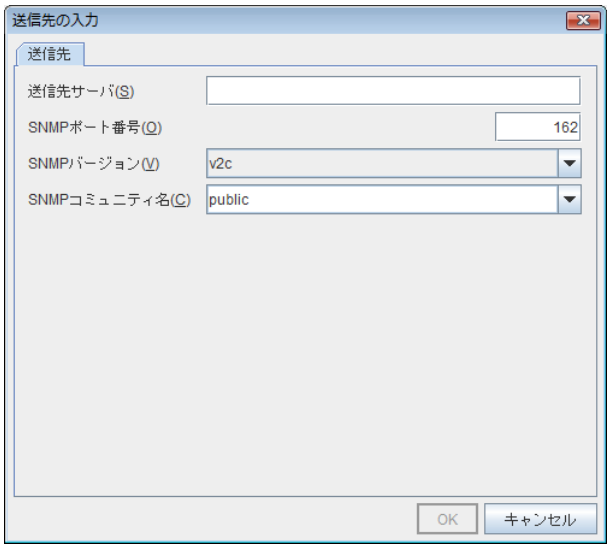
SNMP トラップ送信先を追加します。[追加] をクリックすると 送信先の入力ダイアログが表示されます。

削除

SNMP トラップ送信先の設定を削除する場合に使用します。

編集

SNMP トラップ送信先の設定を変更する場合に使用します。



送信先サーバ (255 バイト以内)

SNMP トラップ送信先のサーバ名を設定します。

SNMP ポート番号 (1-65535)

SNMP トラップ送信先のポート番号を設定します。

SNMP バージョン

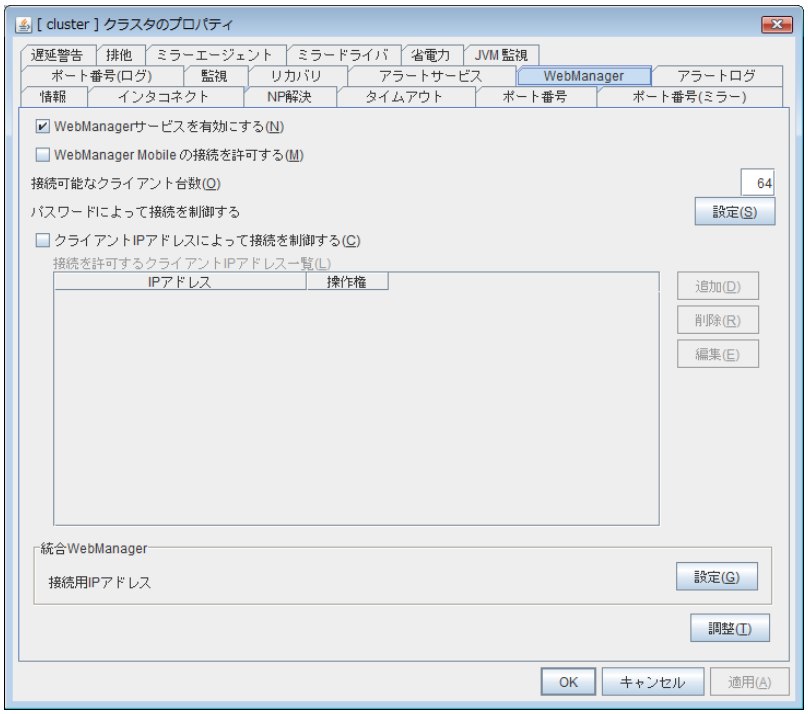
SNMP トラップ送信先の SNMP バージョンを設定をします。

SNMP コミュニティ名 (255 バイト以内)

SNMP トラップ送信先の SNMP コミュニティ名を設定をします。

WebManagerタブ

WebManager を設定します。



WebManager サービスを有効にする

WebManager サービスを有効にします。

- チェックボックスがオン

  WebManager サービスを有効にします。

- チェックボックスがオフ

  WebManager サービスを無効にします。

WebManager Mobile の接続を許可する

WebManager Mobile を有効にします。

- チェックボックスがオン

  WebManager Mobile を有効にします。

- チェックボックスがオフ

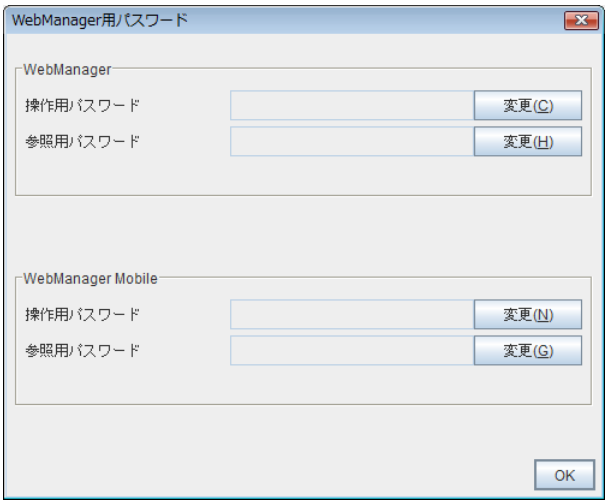
  WebManager Mobile を無効にします。

接続可能なクライアント台数(1～999)

接続可能なクライアント台数を設定します。

パスワードによって接続を制御する

[設定]ボタンを選択すると WebManager 用パスワードダイアログ ボックスが表示されます。



WebManager

- 操作用パスワード

  WebManager に操作モード、設定モード、検証モードで接続するためのパスワードを設定します。
  [変更] をクリックすると [パスワードの変更] ダイアログボックスが表示されます。

- 参照用パスワード

  WebManager に参照モードで接続するためのパスワードを設定します。
  [変更] をクリックすると [パスワードの変更] ダイアログボックスが表示されます。

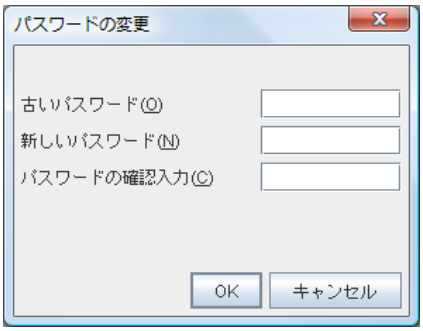
WebManager Mobile

- 操作用パスワード

  WebManager Mobile に操作モードで接続するためのパスワードを設定します。
  [変更] をクリックすると [パスワードの変更] ダイアログボックスが表示されます。

- 参照用パスワード

  WebManager Mobile に参照モードで接続するためのパスワードを設定します。
  [変更] をクリックすると [パスワードの変更] ダイアログボックスが表示されます。



- 古いパスワード(255 バイト以内)
  変更前のパスワードを入力します。
  古いパスワードが設定されていない場合は何も入力しません。
- 新しいパスワード(255 バイト以内)
  新しいパスワードを入力します。
  パスワードを削除する場合は何も入力しません。
- パスワードの確認入力(255 バイト以内)
  新しいパスワードをもう一度入力します。

クライアント IP アドレスによって接続を制御する

クライアント IP アドレスによって接続を制御します。

◆ チェックボックスがオン

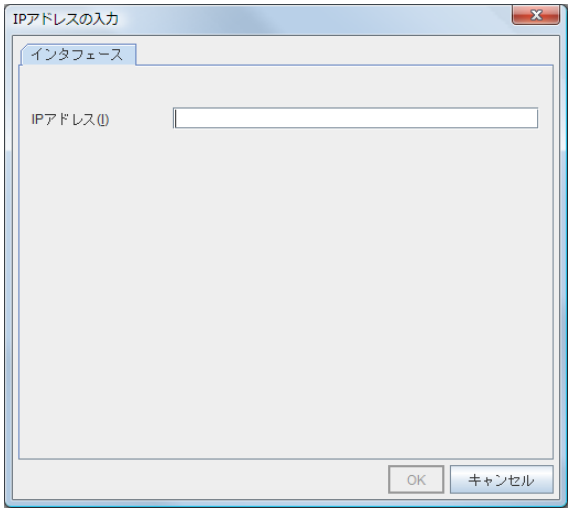
[追加]、[削除]、[編集]ボタンが有効になります。

◆ チェックボックスがオフ

[追加]、[削除]、[編集]ボタンが無効になります。

追加

[接続を許可するクライアント IP アドレス一覧] に IP アドレスを追加する場合に使用します。[追加]ボタンを選択すると IP アドレスの入力ダイアログ ボックスが表示されます。新規に追加する IP アドレスは操作権ありで追加されます。



◆ IP アドレス(80 バイト以内)

接続を許可するクライアント IP アドレスを入力します。

- IP アドレスの場合の例 : 10.0.0.21
- ネットワークアドレスの場合の例 : 10.0.1.0/24

削除

[接続を許可するクライアント IP アドレス一覧] から IP アドレスを削除する場合に使用します。[接続を許可するクライアント IP アドレス一覧] から削除したい IP アドレスを選択して、[削除]ボタンを選択してください。

編集

IP アドレスを編集する場合に使用します。[接続を許可するクライアント IP アドレス一覧] から編集したい IP アドレスを選択して、[編集]ボタンを選択します。選択された IP アドレスが入力されている IP アドレスの入力ダイアログ ボックスが表示されます。編集した IP アドレスの操作権は変わりません。

注: この接続を許可するクライアント IP アドレスは clprexec による外部操作に対する接続制限にも使用されます。

操作権

[接続を許可するクライアント IP アドレス一覧] に登録されている IP アドレスに操作権を設定します。

- チェックボックスがオン

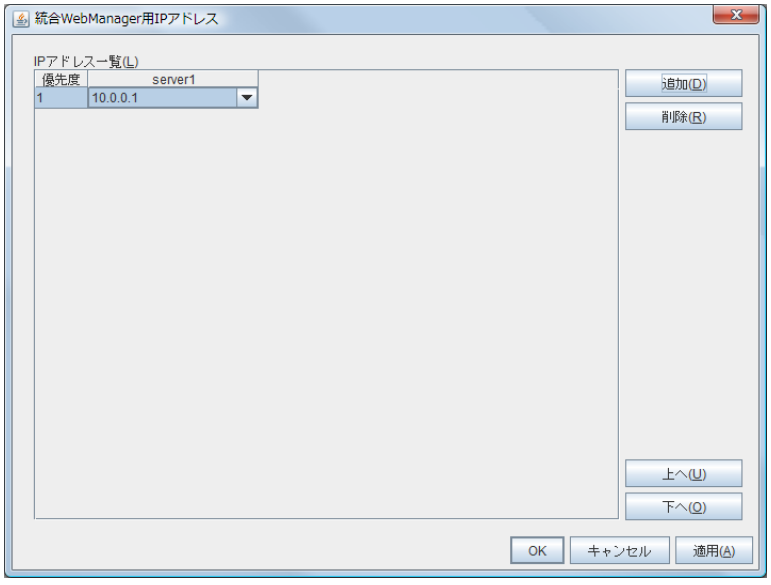
  クライアントは CLUSTERPRO X SingleServerSafe の操作と状態表示が行えます。

- チェックボックスがオフ

  クライアントは CLUSTERPRO X SingleServerSafe の状態表示のみ行えます。

接続用 IP アドレス

[設定] ボタンをクリックすると 統合 WebManager 用 IP アドレスダイアログ ボックスが表示されます。



- 追加

  統合 WebManager 用 IP アドレスを追加します。各サーバの IP アドレスは、各サーバの列のセルをクリックして IP アドレスを選択または入力して設定します。

- 削除

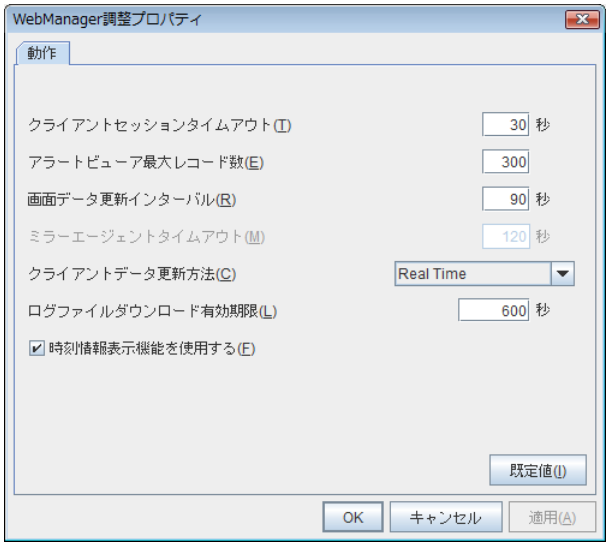
  通信経路を削除します。削除したい通信経路の列を選択して [削除] をクリックすると、選択していた経路が削除されます

- 上へ、下へ

統合 WebManager 用 IP アドレスを複数設定する場合、[優先度] 列の番号が小さい通信経路が優先的にクラスタ サーバ間の内部通信に使用されます。優先度を変更する場合は、[上へ] [下へ] をクリックして、選択行の順位を変更します。

調整

WebManager の調整を行う場合に使用します。[調整]ボタンを選択すると [WebManager 調整プロパティ] ダイアログ ボックスが表示されます。



- クライアントセッションタイムアウト(1～999)

  WebManager サーバが WebManager と通信しなくなってからのタイムアウト時間です。

- アラートビューア最大レコード数(1～999)

  WebManager のアラートビューアに表示される最大のレコード数です。

- 画面データ更新インターバル(0～999)

  WebManager の画面データが更新される間隔です。

- ミラーエージェントタイムアウト（1～999）

  ミラーディスクの情報取得処理のタイムアウト時間です。

- クライアントデータ更新方法

  WebManager の画面データの更新方法を下記より選択できます。

  - Polling
    画面データは定期的に更新されます。
  - Real Time
    画面データはリアルタイムに更新されます。

- ログファイルダウンロード有効期限(60～43200)

  サーバ上に一時保存したログ収集情報を削除するまでの有効期限です。ログ収集情報の保存ダイアログが表示されてから、保存を実行しないまま有効期限が経過するとサーバ上のログ収集情報は削除されます。

◆ 時刻情報表示機能を使用する

時刻情報表示機能の有効/無効を設定します。

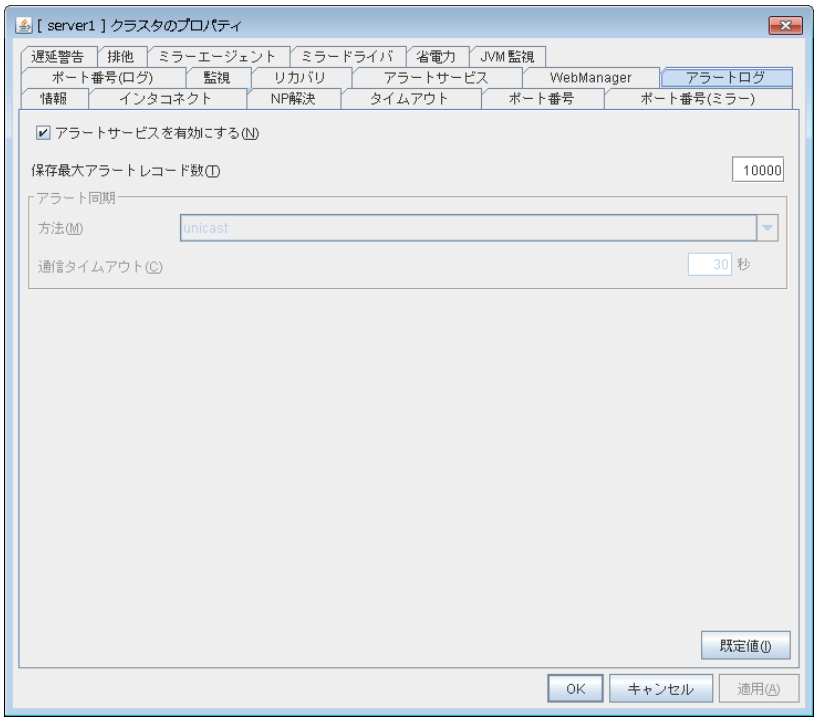
- チェックボックスがオン
  時刻情報表示機能を有効にします。
- チェックボックスがオフ
  時刻情報表示機能を無効にします。

◆ 既定値

既定値に戻すときに使用します。[既定値]ボタンを選択すると全ての項目に既定値が設定されます。

アラートログタブ

アラートログを設定します。



アラートサービスを有効にする

サーバのアラートサービスを起動するかどうかの設定です。

- チェックボックスがオン

  アラートサービスを有効にします。

- チェックボックスがオフ

  アラートサービスを無効にします。

保存最大アラートレコード数(1～99999)

サーバのアラートサービスが保存できる最大のアラートメッセージ数です。

アラート同期　方法

使用しません。

アラート同期　通信タイムアウト(1～300)

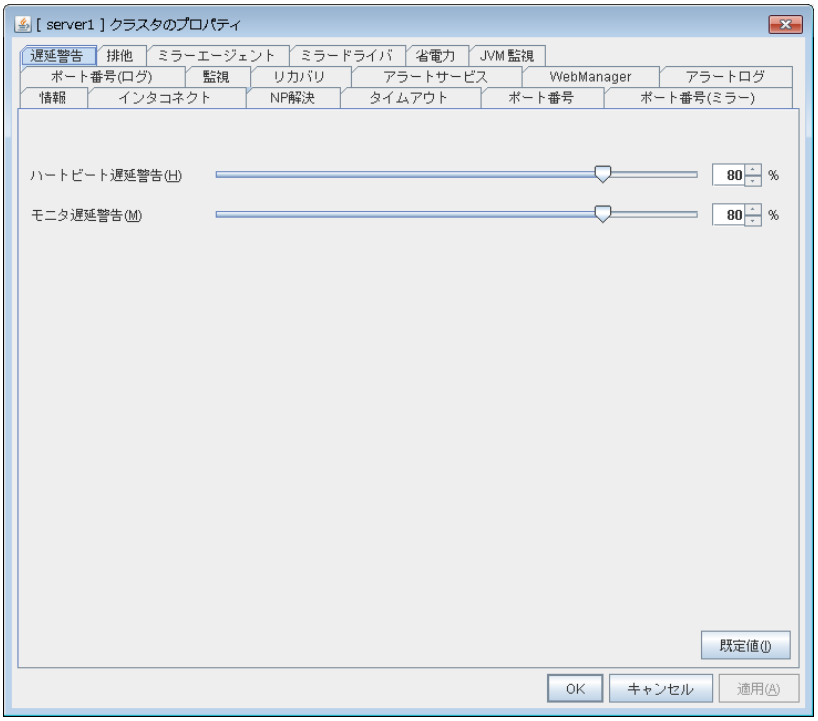
使用しません。

既定値

既定値に戻すときに使用します。[既定値]ボタンを選択すると全ての項目に既定値が設定されます。

遅延警告タブ

遅延警告を設定します。遅延警告の詳細については「第 8 章 監視動作の詳細」の「モニタリソースの遅延警告」を参照してください。



ハートビート遅延警告(0～100)

ハートビートの遅延警告の割合を設定します。ハートビートタイムアウト時間のここで指定した割合の時間内にハートビートの応答がない場合にアラートログに警告を表示します。100 を設定すると警告を表示しません。

モニタ遅延警告(0～100)

モニタの遅延警告の割合を設定します。モニタタイムアウト時間のここで指定した割合の時間内にモニタの応答がない場合にアラートログに警告を表示します。100 を設定すると警告を表示しません。

注: 遅延警告で 0%を指定するとハートビートインターバル、モニタインターバルごとにアラートログを表示します。

アラートログで監視にかかった時間を確認することができるので、テスト運用などで監視の時間を確認する場合は、0%を設定します。

本番環境では 0%などの低い値は設定しないでください。

排他タブ

使用しません。

ミラーエージェントタブ

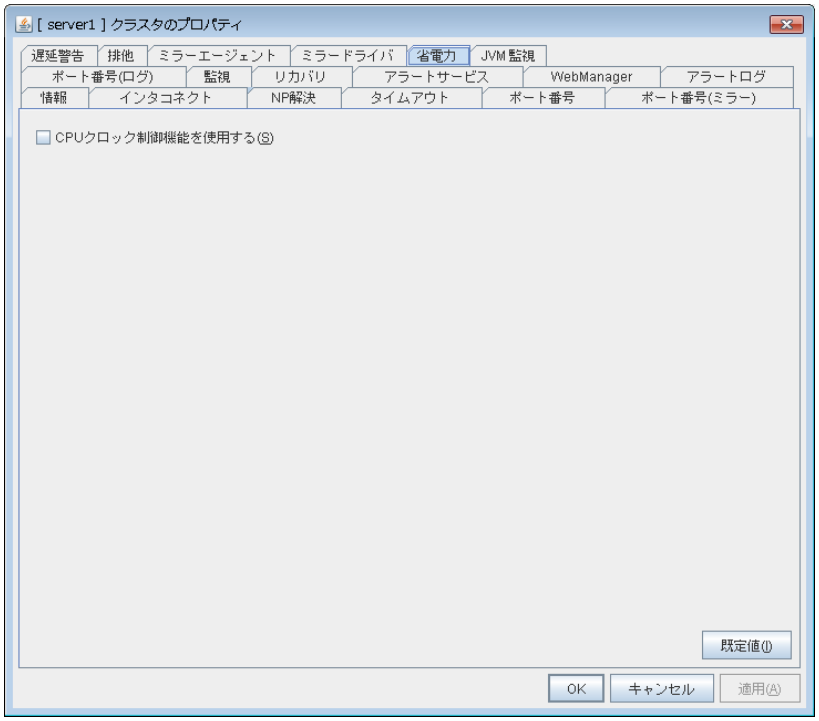
使用しません。

ミラードライバタブ

使用しません。

省電力タブ

待機系サーバの CPU クロックを制御して省電力モードにする機能を使用するかどうかを設定します。



CPU クロック制御機能を使用する

CPU クロック制御の設定をします。

チェックボックスをオンに設定するとグループ起動時に CPU クロック数を最高 (high) に、グループ停止後にサーバのクロック数を最低 (low) に設定します。

チェックボックスをオフに設定すると CPU クロック制御機能は動作しません。

コマンドや WebManager で CPU クロック制御を行った場合は、グループの起動/停止に関わらず、コマンドや WebManager で変更された設定が優先されます。ただし、サーバの停止/起動またはサスペンド/リジューム後には、コマンドや WebManager で変更された設定は破棄され、CPU クロックはサーバから制御されます。

- チェックボックスがオン

  CPU クロック制御を行います。

- チェックボックスがオフ

  CPU クロック制御を行いません。

既定値

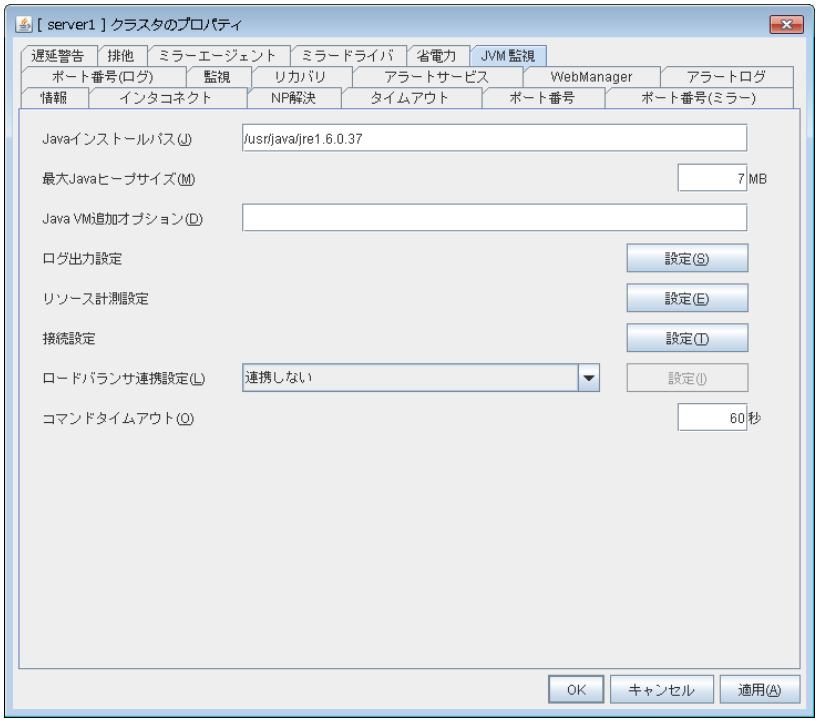
既定値に戻すときに使用します。[既定値]ボタンを選択すると全ての項目に既定値が設定されます。

注:CPU クロック制御機能を使用する場合、BIOS の設定でクロックの変更が可能になっていることと、CPU が OS の電源管理機能によるクロック制御をサポートしていること、カーネルが対応していることが必要となります。

JVM監視タブ

JVM 監視で用いる詳細なパラメータを設定します。

注: オンライン版 Builder で JVM 監視タブを表示するためには、Java Resource Agent のライセンスが登録されている状態で[ファイル]メニューの[サーバ情報の更新]を実行する必要があります。



Java インストールパス(255 バイト以内)

JVM 監視が使用する Java VM のインストールパスを設定します。絶対パスかつ ASCII 文字で指定してください。末尾に"/"はつけないでください。クラスタ内のサーバにおいて、共通の設定となります。指定例:/usr/java/jre1.6.0_37

最大 Java ヒープサイズ(7～4096)

JVM 監視が使用する Java VM の最大ヒープサイズをメガバイトで設定します(Java VM 起動時オプションの-Xmx に相当)。クラスタ内のサーバにおいて、共通の設定となります。Oracle Java の場合は 7 以上、JRockit の場合は 16 以上を指定してください。

Java VM 追加オプション(1024 バイト以内)

JVM 監視が使用する Java VM の起動時オプションを設定します。ただし、-Xmx は[最大 Java ヒープサイズ]で指定してください。クラスタ内のサーバにおいて、共通の設定となります。指定例:-XX:+UseSerialGC

ログ出力設定

[設定]ボタンを押すとログ出力設定入力のダイアログが表示されます。

リソース計測設定

[設定]ボタンを押すとリソース計測設定入力のダイアログが表示されます。

接続設定

[設定]ボタンを押すと接続設定入力のダイアログが表示されます。

ロードバランサ連携設定

ロードバランサ種別を選択し、[設定]ボタンを押すとロードバランサ連携設定入力のダイアログが表示されます。

ロードバランサ種別は、リストから選択します。ロードバランサ連携する場合は、ご利用のロードバランサを選択してください。ロードバランサ連携しない場合は、[連携しない]を選択してください。

コマンドタイムアウト(30～300)

JVM 監視の各画面で指定する[コマンド]のタイムアウト値を設定します。[コマンド]共通の設定となります。

ログ出力設定

[設定]ボタンを押すとログ出力設定入力のダイアログが表示されます。

ログレベル

JVM 監視が出力するログのログレベルを選択します。

保持する世代数(2～100)

JVM 監視が出力するログについて保持する世代数を設定します。

ローテーション方式

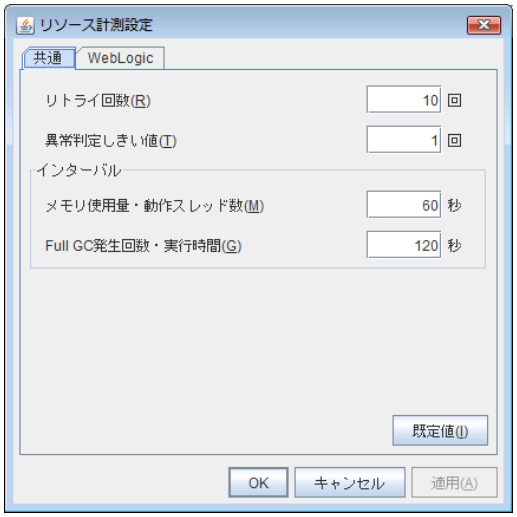
JVM 監視が出力するログのローテーション方式を選択します。ファイルサイズによるログローテーションの場合、JVM 運用ログなどログ 1 ファイルあたりの最大サイズをキロバイトで設定します(範囲は 200～2097151)。時間によるログローテーションの場合、ログローテーション開始時刻を”hh:mm”の形式(hh:時間を 0～23、mm:分を 0～59 で指定)、ローテーションのインターバルを時間(範囲は 1～8784)で設定します。

既定値

ログレベル、保持する世代数、ローテーション方式を既定値の設定に戻します。

リソース計測設定[共通]

[設定]ボタンを押すとリソース計測設定入力のダイアログが表示されます。JVM 監視が異常と判断する仕組みについては、「第 5 章 モニタリソースの詳細」を参照してください。



リトライ回数(1～1440)

JVM 監視がリソース計測失敗した時のリソース計測リトライ回数を設定します。

異常判定しきい値(1～10)

JVM 監視がリソース計測により取得した Java VM やアプリケーションサーバのリソース使用量が、お客様定義のしきい値を連続して超えた場合に異常と判断する回数を設定します。

インターバル メモリ使用量・動作スレッド数(15～600)

JVM 監視がメモリ使用量および動作スレッド数を計測するインターバルを設定します。

インターバル Full GC 発生回数・実行時間(15～600)

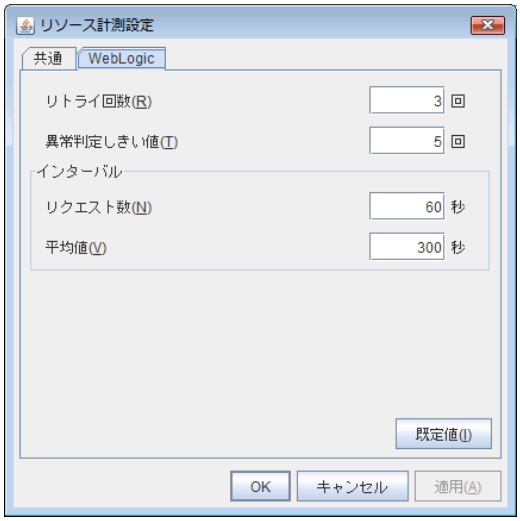
JVM 監視が Full GC 発生回数および発生時間を計測するインターバルを設定します。

既定値

リトライ回数、異常判定しきい値、インターバルを既定値の設定に戻します。

リソース計測設定[WebLogic]

[設定]ボタンを押すとリソース計測設定入力のダイアログが表示されます。JVM 監視が異常と判断する仕組みについては、「第 5 章モニタリソースの詳細」を参照してください。



リトライ回数(1～5)

JVM 監視がリソース計測失敗した時のリソース計測リトライ回数を設定します。

異常判定しきい値(1～10)

JVM 監視がリソース計測により取得した Java VM やアプリケーションサーバのリソース使用量が、お客様定義のしきい値を連続して超えた場合に異常と判断する回数を設定します。

インターバル リクエスト数(15～600)

JVM 監視が WebLogic 監視におけるワークマネージャおよびスレッドプールのリクエスト数を計測するインターバルを設定します。

インターバル 平均値(15～600)

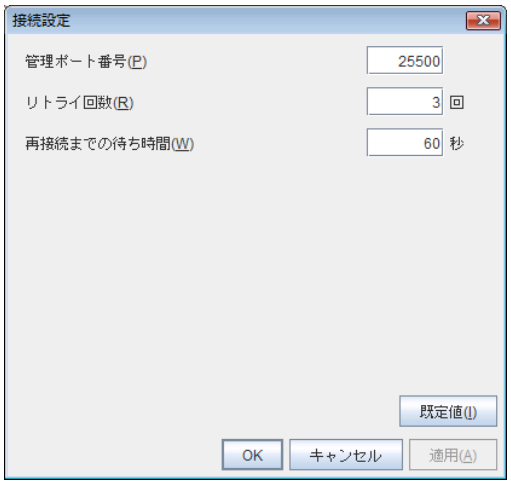
JVM 監視が WebLogic 監視におけるワークマネージャおよびスレッドプールのリクエスト数平均値を計測するインターバルを設定します。インターバル リクエスト数で設定されている整数倍の値を設定してください。

既定値

リトライ回数、異常判定しきい値、インターバルを既定値の設定に戻します。

接続設定

[設定]ボタンを押すと監視対象の Java VM への接続設定入力のダイアログが表示されます。



管理ポート番号(10000～65535)

監視対象の Java VM に接続するためのポート番号を設定します。クラスタ内のサーバにおいて、共通の設定となります。32768～61000 は非推奨です。

リトライ回数(1～5)

監視対象の Java VM へ接続失敗時のリトライ回数を設定します。

再接続までの待ち時間(15～60)

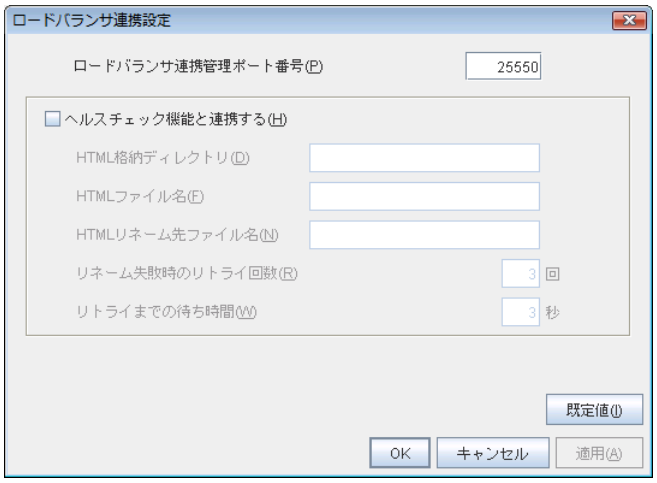
監視対象の Java VM へ接続失敗時に接続をリトライするまでのインターバルを設定します。

既定値

管理ポート番号、リトライ回数、再接続までの待ち時間を既定値の設定に戻します。

ロードバランサ連携設定

ロードバランサ種別として[BIG-IP LTM]以外を選択し、[設定]ボタンを押すとロードバランサ連携設定入力のダイアログが表示されます。



ロードバランサ連携管理ポート番号(10000～65535)

ロードバランサ連携機能にて使用するためのポート番号を設定します。クラスタ内のサーバにおいて、共通の設定となります。32768～61000 は非推奨です。

ヘルスチェック機能と連携する

監視対象の Java VM の障害検出時、ロードバランサによるヘルスチェック機能を使用するかを設定します。

HTML 格納ディレクトリ(1023 バイト以内)

ロードバランサによるヘルスチェック機能が使用する HTML ファイルが格納されているディレクトリを設定します。絶対パスかつ ASCII 文字で指定してください。末尾に"/"はつけないでください。

HTML ファイル名(255 バイト以内)

ロードバランサによるヘルスチェック機能が使用する HTML ファイル名を設定します。ASCII 文字で指定してください。

HTML リネーム先ファイル名(255 バイト以内)

ロードバランサによるヘルスチェック機能が使用する HTML リネーム先ファイル名を設定します。ASCII 文字で指定してください。[HTML ファイル名]とは異なるファイル名を指定してください。

リネーム失敗時のリトライ回数(0～5)

HTML ファイルのリネーム失敗時にリネームをリトライする回数を設定します。

リトライまでの待ち時間(1～60)

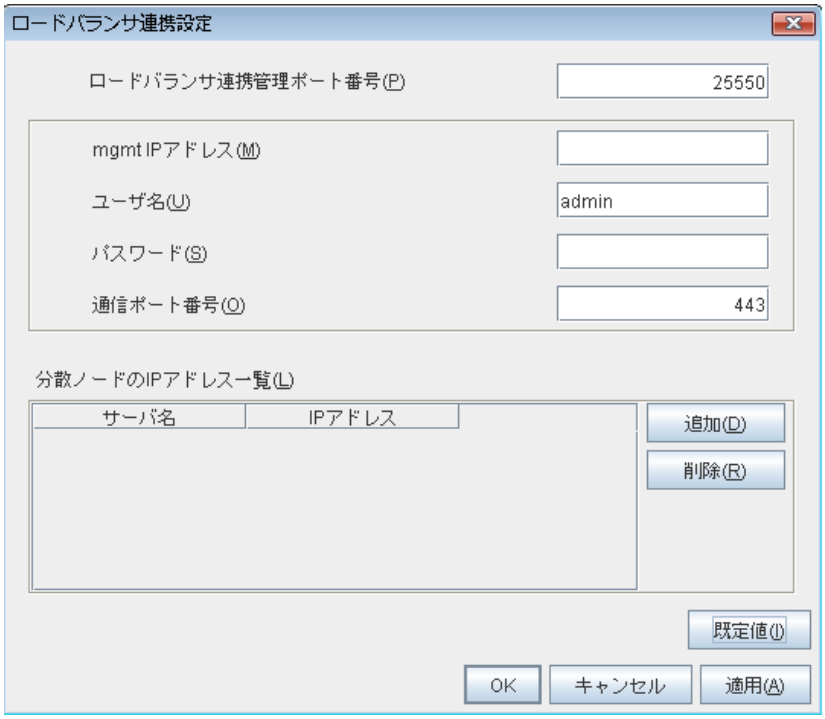
HTML ファイルのリネーム失敗時にリネームをリトライするまでのインターバルを設定します。

既定値

ロードバランサ連携管理ポート番号、ヘルスチェック機能と連携する、HTML 格納ディレクトリ、HTML ファイル名、HTML リネーム先ファイル名、リネーム失敗時のリトライ回数、リトライまでの待ち時間を既定値の設定に戻します。

ロードバランサ連携設定

ロードバランサ種別として[BIG-IP LTM]を選択し、[設定]ボタンを押すとロードバランサ連携設定入力のダイアログが表示されます。



ロードバランサ連携管理ポート番号(10000～65535)

ロードバランサ連携機能にて使用するためのポート番号を設定します。クラスタ内のサーバにおいて、共通の設定となります。42424～61000 は非推奨です。

mgmt IP アドレス

BIG-IP LTM の IP アドレスを設定します。

管理ユーザ名(255 バイト以内)

BIG-IP LTM の管理ユーザ名を設定します。

パスワード(255 バイト以内)

BIG-IP LTM の管理ユーザパスワードを設定します。

通信ポート番号(10000～65535)

BIG-IP LTM との通信用ポート番号を設定します。

追加

分散ノードのサーバ名と IP アドレスを追加します。サーバ名は CLUSTERPRO のサーバ名、IP アドレスは BIG-IP Configration Utility の [LocalTrafic]-[Pools:PoolList]-[該当の pool]-[Members] の Members と同じ値を設定してください。変更する場合は、変更したい行を選択して、直接編集してください。

削除

分散ノードのサーバ名とIPアドレスを削除します。削除したい行を選択して、[削除]をクリックすると、選択していたサーバが削除されます。

既定値

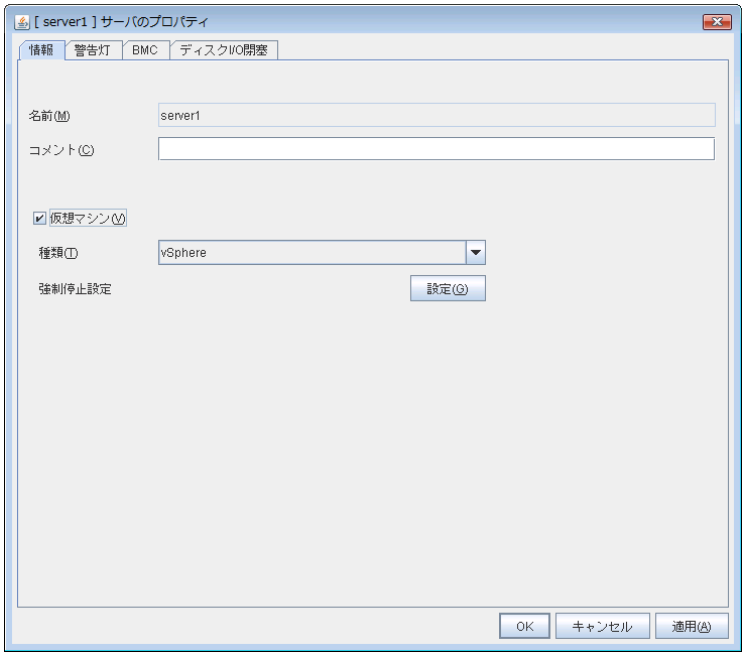
ロードバランサ連携管理ポート番号、管理ユーザ名、通信ポート番号を既定値の設定に戻します。

サーバプロパティ

「サーバのプロパティ」では、サーバ固有の設定を行います。

情報タブ

サーバ名の表示、コメントの登録、変更を行います。



名前

サーバ名を表示しています。ここでは名前の変更はできません。

コメント(127 バイト以内)

サーバのコメントを設定します。半角英数字のみ入力可能です。

仮想マシン

このサーバが仮想マシン（ゲスト OS）であるかどうかを指定します。

- チェックボックスがオン

  仮想マシン（ゲスト OS）であることを示します。仮想マシンの設定が可能になります。

- チェックボックスがオフ

  物理マシンであることを示します。仮想マシンの設定はできません。

種類

仮想化基盤の種類を指定します。

- vSphere
  VMware 社の仮想化基盤です。

- KVM
  Linux カーネル仮想化基盤です。
- XenServer
  Citrix 社の仮想化基盤です。
- Hyper-V
  Microsoft 社の仮想化基盤です。
- other
  その他の仮想化基盤を使用する場合に指定します。

強制停止設定

使用しません。

警告灯タブ

使用しません。

BMCタブ

使用しません。

ディスク I/O 閉塞タブ

使用しません。

セクション IV 監視のしくみ

このセクションでは、CLUSTERPRO X SingleServerSafe の監視のしくみについての詳細を説明します。

第 8 章　　監視動作の詳細

第 8 章　　監視動作の詳細

本章では、監視における監視インターバル、監視タイムアウト、監視リトライ回数をどうのように設定すればよいか検討するために、いくつかの障害パターンにより、どのように障害を検出するかの仕組みについての詳細を説明します。

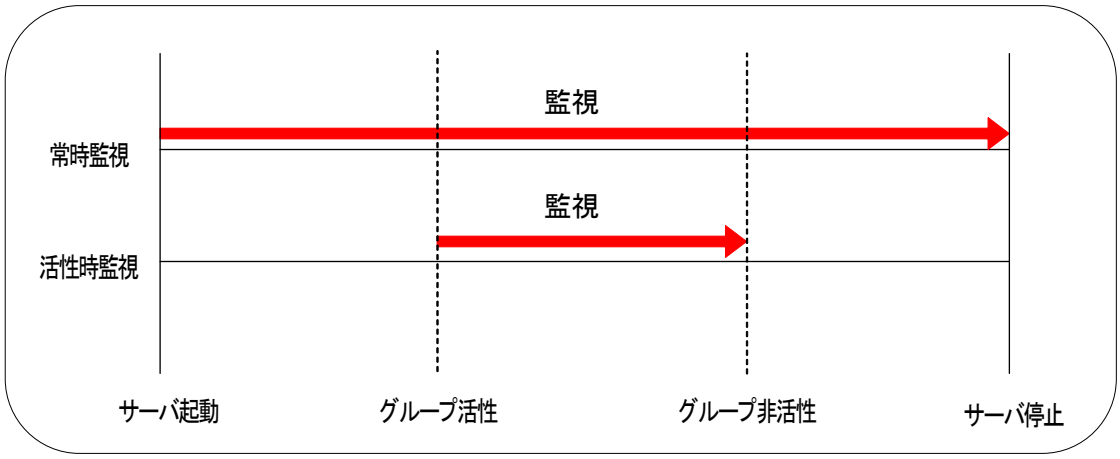
本章で説明する項目は以下のとおりです。

常時監視と活性時監視について

常時監視では、サーバが起動して、CLUSTERPRO X SingleServerSafe が動作可能になった時点から監視を始めます。

活性時監視では、指定されたグループが活性してから、そのグループが非活性(停止)する間で監視が行われます。

モニタリソースにより、いずれかに固定されているもの、いずれかを選択できるものがあります。



モニタリソースの監視インターバルのしくみ

全てのモニタリソースは、監視インターバル毎に監視が行われます。

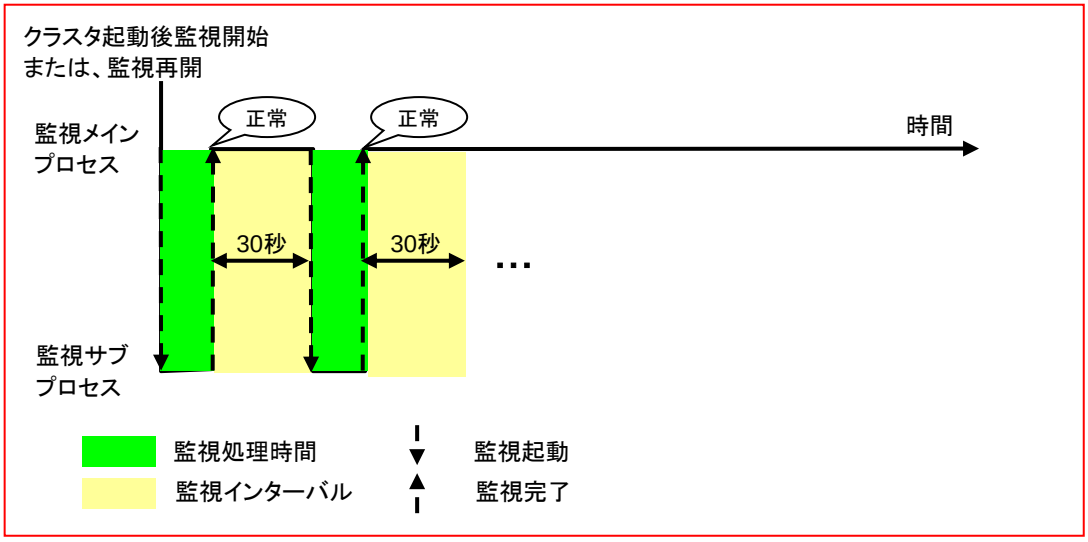
以下は、この監視インターバルの設定による正常または、異常時におけるモニタリソースの監視の流れを時系列で表した説明です。

監視正常検出時

下記の値が設定されている場合の挙動の例:

<監視>

監視インターバル	30 秒
監視タイムアウト	60 秒
監視リトライ回数	0 回



監視異常検出時(監視リトライ設定なし)

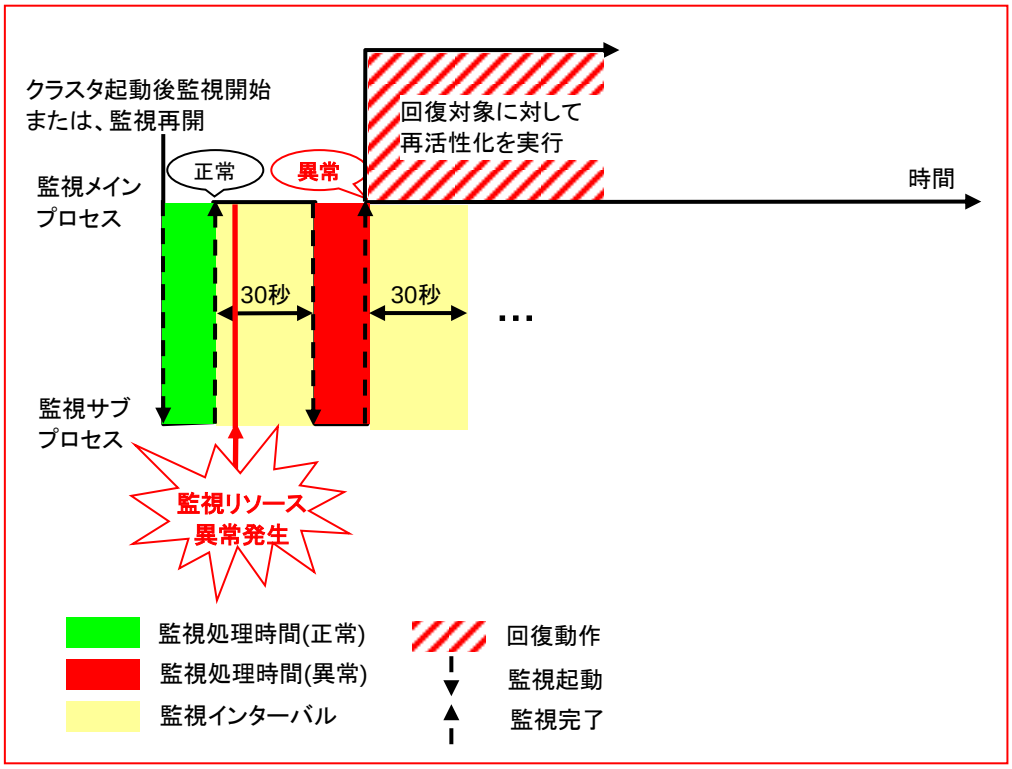
下記の値が設定されている場合の挙動の例:

<監視>

監視インターバル	30 秒
監視タイムアウト	60 秒
監視リトライ回数	0 回

<異常検出>

回復動作	回復対象を再起動
回復対象	グループ
回復スクリプト実行回数	0 回
再活性化しきい値	1 回
最終動作	何もしない



監視異常発生後、次回監視で監視異常を検出し回復対象に対して再活性化が行われます。

監視異常検出時(監視リトライ設定あり)

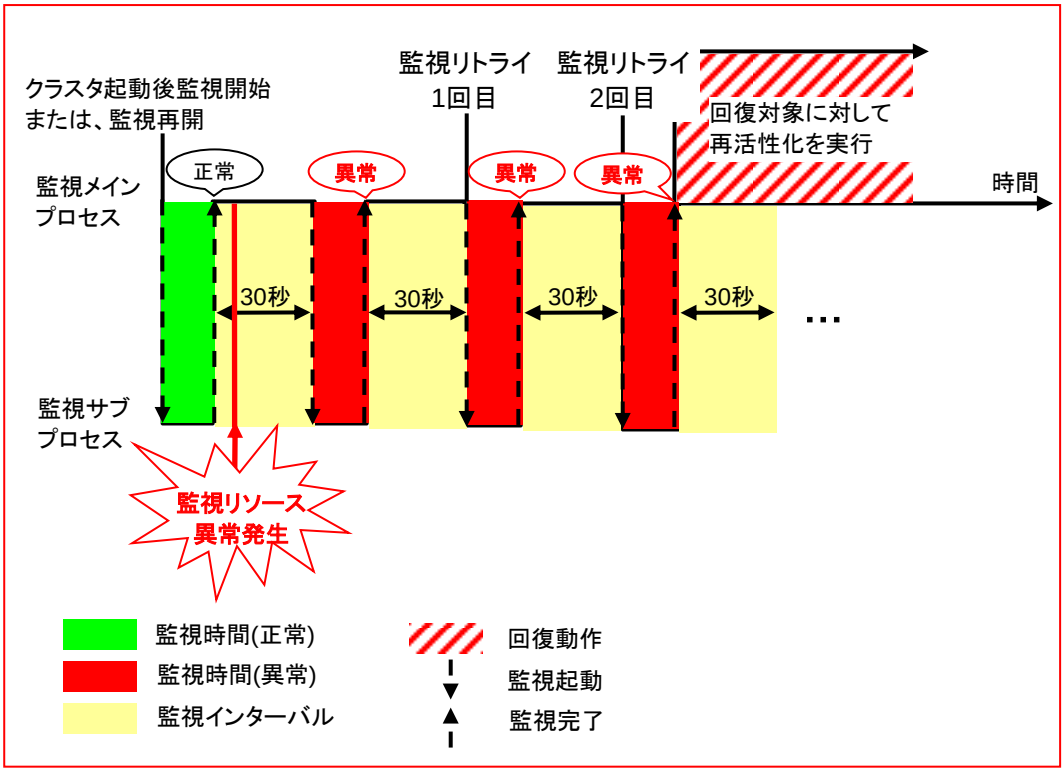
下記の値が設定されている場合の挙動の例:

<監視>

監視インターバル	30 秒
監視タイムアウト	60 秒
監視リトライ回数	2 回

<異常検出>

回復動作	回復対象を再起動
回復対象	グループ
回復スクリプト実行回数	0 回
再活性化しきい値	1 回
最終動作	何もしない



監視異常発生後、次回監視で監視異常を検出し監視リトライ以内で回復しなければ、回復対象に対して再活性化が行われます。

監視タイムアウト検出時(監視リトライ設定なし)

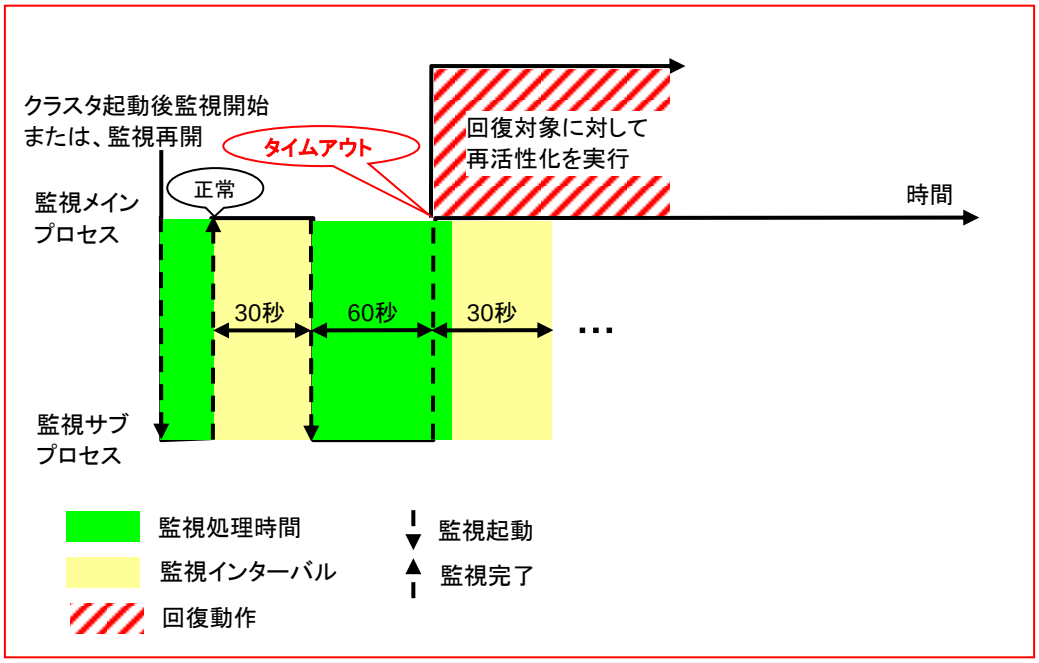
下記の値が設定されている場合の挙動の例:

<監視>

監視インターバル	30 秒
監視タイムアウト	60 秒
監視リトライ回数	0 回

<異常検出>

回復動作	回復対象を再起動
回復対象	グループ
回復スクリプト実行回数	0 回
再活性化しきい値	1 回
最終動作	何もしない



監視タイムアウト発生後、直ぐに回復対象への回復動作に対して再活性化が行われます。

監視タイムアウト検出時(監視リトライ設定あり)

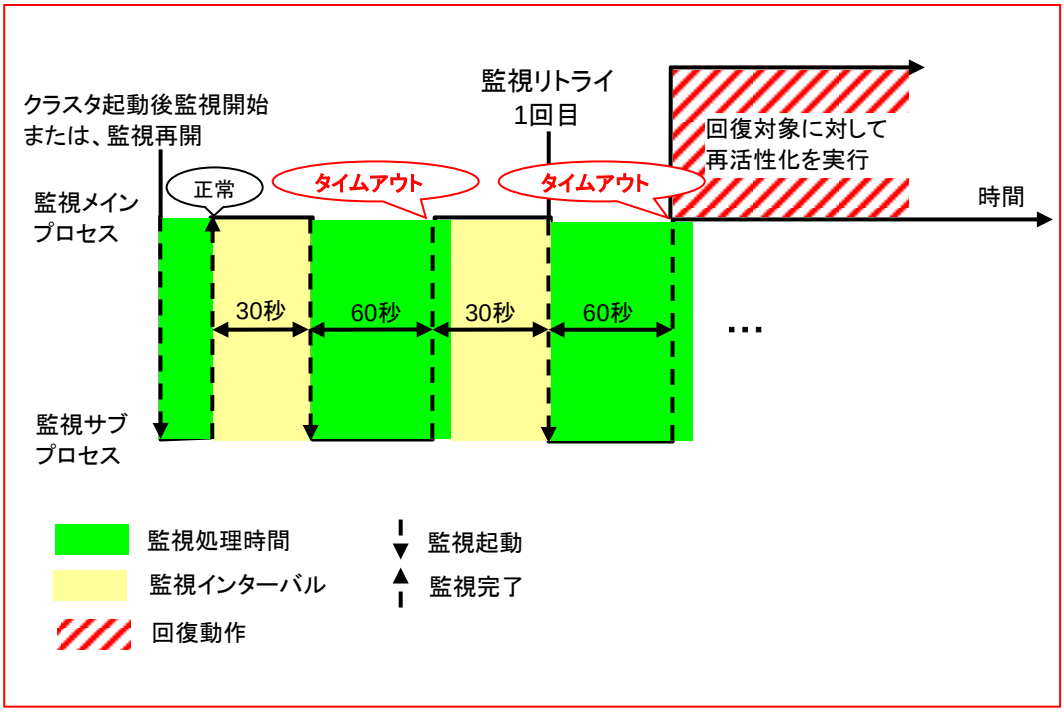
下記の値が設定されている場合の挙動の例:

<監視>

監視インターバル	30 秒
監視タイムアウト	60 秒
監視リトライ回数	1 回

<異常検出>

回復動作	回復対象を再起動
回復対象	グループ
回復スクリプト実行回数	0 回
再活性化しきい値	1 回
最終動作	何もしない



監視タイムアウト発生後、監視リトライを行い回復対象に対して再活性化が行われます。

モニタリソースによる異常検出時の動作

異常検出時には回復対象に対して以下の回復動作が行われます。

- 監視対象の異常を検出すると回復スクリプトを実行します。
- 回復スクリプト実行回数の回復スクリプト実行後、回復対象の再活性化を行います。再活性前スクリプト実行が設定されている場合はスクリプトを実行後に再活性化を行います。
- 監視対象の異常を検出すると回復対象の再活性化を行います（回復動作が [最終動作のみ実行] の場合、及び [カスタム設定] で最大再活性回数が0に設定されている場合は再活性化を行いません）。
- 再活性化に失敗した場合、あるいは再活性化を行っても異常を検出する場合、最終動作を行います（[カスタム設定] で最大再活性回数が2以上に設定されている場合は、指定回数まで再活性化をリトライします）。

回復動作は、回復対象が以下の状態であれば行われません。

回復対象	状態	再活性化[5]	最終動作[6]
グループ/ グループリソース	停止済	×	×
	起動/停止中	×	×
	起動済	○	○
	異常	○	○
LocalServer	-	-	○

注: モニタリソースの異常検出時の設定で回復対象にグループリソース(例: EXEC リソース、仮想マシンリソース)を指定し、モニタリソースが異常を検出した場合の回復動作遷移中(再活性化 → 最終動作)には、以下のコマンドまたは WebManager から以下の操作を行わないでください。

- サーバの停止 / サスペンド
- グループの開始 / 停止

モニタリソース異常による回復動作遷移中に上記の制御を行うと、そのグループの他のグループリソースが停止しないことがあります。

また、モニタリソース異常状態であっても最終動作実行後であれば上記制御を行うことが可能です。

モニタリソースの状態が異常から復帰(正常)した場合は、再活性化回数、最終動作の実行要否はリセットされます。ただし、回復対象としてグループ/グループリソースが指定されている場合は、同一の回復対象が指定されている全てのモニタリソースの状態が正常状態になった場合のみ、これらのカウンタがリセットされます。

回復動作の再活性化回数は、回復動作に失敗した場合でも 1 回としてカウントされることに注意してください。

[5] 再活性化しきい値に1以上が設定されている場合のみ有効になります。
[6] 最終動作に"何もしない"以外が設定されている場合のみ有効になります。

監視異常からの復帰(正常)

監視異常を検出し、回復動作遷移中または全ての回復動作を完了後にモニタリソースの復帰を検出すると、そのモニタリソースが保持している以下のしきい値に対する回数カウンタはリセットされます。

- 回復スクリプト実行回数
- 再活性化回数

最終動作については、実行要否がリセットされます。

回復動作時の回復対象活性/非活性異常

モニタリソースの監視先と回復対象のグループリソースが同一のデバイスの場合で監視異常を検出すると、回復動作中にグループリソースの活性/非活性異常を検出する場合があります。

回復スクリプト、回復動作前スクリプトについて

モニタリソースの異常検出時に、回復スクリプトを実行させることが可能です。また、回復対象の再活性化、最終動作を実行する前に回復動作前スクリプトを実行させることも可能です。

いずれの場合でも共通のスクリプトファイルが実行されます。

回復スクリプト、回復動作前スクリプトで使用する環境変数

CLUSTERPRO はスクリプトを実行する場合に、どの状態で実行したか (回復動作種別) などの情報を環境変数にセットします。

スクリプト内で下図の環境変数を分岐条件として、システム運用にあった処理内容を記述できます。

環境変数	環境変数の値	意味
CLP_MONITORNAME …モニタリソース名	モニタリソース名	回復スクリプト、回復動作前スクリプトを実行する原因となる異常を検出したモニタリソース名を示します。
CLP_VERSION_FULL …CLUSTERPROフルバージョン	CLUSTERPROフルバージョン	CLUSTERPROのフルバージョンを示す。 (例)3.3.0-1
CLP_VERSION_MAJOR …CLUSTERPROメジャーバージョン	CLUSTERPROメジャーバージョン	CLUSTERPROのメジャーバージョンを示す。 (例)3
CLP_PATH …CLUSTERPROインストールパス	CLUSTERPROインストールパス	CLUSTERPROがインストールされているパスを示す。 (例)/opt/nec/clusterpro
CLP_OSNAME …サーバOS名	サーバOS名	スクリプトが実行されたサーバのOS名を示す。 (例) ①OS名が取得できた場合: Red Hat Enterprise Linux Server release 6.0 (Santiago) ②OS名が取得できなかった場合: Linux
CLP_OSVER …サーバOSバージョン	サーバOSバージョン	スクリプトが実行されたサーバのOSバージョンを示す。 (例) ①OSバージョンが取得できた場合:6.0 ②OSバージョンが取得できなかった場合:※値なし
CLP_ACTION …回復動作種別	RECOVERY	回復スクリプトとして実行された場合。
	RESTART	再起動前に実行された場合。

環境変数	環境変数の値	意味
	FINALACTION	最終動作前に実行された場合。
CLP_RECOVERYCOUNT …回復スクリプトの実行回数	回復スクリプト実行回数	何回目の回復スクリプト実行回数かを示す。
CLP_RESTARTCOUNT …再活性化回数	再活性化回数	何回目の再活性化回数かを示す。
CLP_FAILOVERCOUNT …フェイルオーバ回数	フェイルオーバ回数	使用しません。

回復スクリプト、回復動作前スクリプトの記述の流れ

前のトピックの、環境変数と実際のスクリプト記述を関連付けて説明します。

回復スクリプト、回復動作前スクリプトの一例

```
#! /bin/sh
# ****************************************
# *                preaction.sh                *
# ****************************************

if [ "$CLP_ACTION" = "RECOVERY" ]
then
    処理概要：
        回復処理
    この処理を行う実行タイミング：
        回復動作 ： 回復スクリプト

elif [ "$CLP_ACTION" = "RESTART" ]
then
    処理概要：
        再活性化前処理
    この処理を行う実行タイミング：
        回復動作 ： 再活性化

elif [ "$CLP_ACTION" = "FINALACTION" ]
then
    処理概要：
        回復処理
    この処理を行う実行タイミング：
        回復動作 ： 最終動作

fi
exit 0
```

スクリプト実行要因の環境変数を参照して、処理の振り分けを行う。

回復スクリプト、回復動作前スクリプト作成のヒント

以下の点に注意して、スクリプトを作成してください。

- スクリプト中にて、実行に時間を必要とするコマンドを実行する場合には、コマンドの実行が完了したことを示すトレースを残すようにしてください。この情報は、問題発生時、障害の切り分けを行う場合に使用することができます。clplogcmdを使用してトレースを残す方法があります。
- スクリプト中に clplogcmd を使用して記述する方法
  clplogcmd で WebManager のアラートビューや OS の syslog に、メッセージを出力できます。clplogcmd については、『操作ガイド』の『第 2 章 CLUSTERPRO X SingleServerSafe コマンドリファレンス メッセージを出力する (clplogcmd コマンド)』を参照してください。

  (例:スクリプト中のイメージ)
  ```
  clplogcmd -m "recoverystart.."
  recoverystart
  clplogcmd -m "OK"
  ```

回復スクリプト、回復動作前スクリプト 注意事項

- スクリプトから起動されるコマンド、アプリケーションのスタックサイズについて

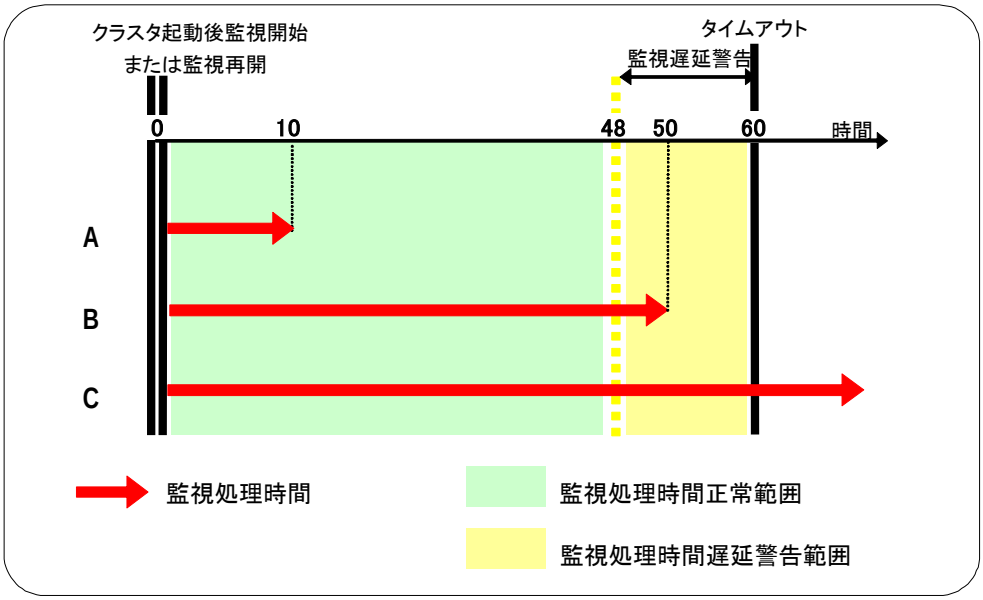
  スタックサイズが 2MB に設定された状態で回復スクリプト、回復動作前スクリプトが実行されます。このため、スクリプトから起動されるコマンドやアプリケーションで 2MB 以上のスタックサイズが必要な場合には、スタックオーバーフローが発生します。
  スタックオーバーフローが発生する場合には、コマンドやアプリケーションを起動する前にスタック サイズを設定してください。

モニタリソースの遅延警告

モニタリソースは、業務アプリケーションの集中などにより、サーバが高負荷状態になり監視タイムアウトを検出する場合があります。監視タイムアウトを検出する前に監視の監視処理時間(実測時間)が監視タイムアウト時間の何割かに達した場合、アラート通報させることが可能です。

以下は、モニタリソースが遅延警告されるまでの流れを時系列で表した説明です。

監視タイムアウトに 60 秒、遅延警告割合には、既定値の 80%を指定します。



A. 監視の監視処理時間は10秒で、モニタリソースは正常状態。
この場合、アラート通報は行いません。

B. 監視の監視処理時間は50秒で、監視の遅延を検出し、モニタリソースは正常状態。
この場合、遅延警告割合の80%を超えているためアラート通報を行います。

C. 監視の監視処理時間は監視タイムアウト時間の60秒を越え、監視タイムアウトを検出し、モニタリソースは異常状態。
この場合、アラート通報は行いません。

また、遅延警告割合を 0 または、100 に設定すれば以下を行うことが可能です。

- 遅延警告割合に0を設定した場合

  監視ごとに遅延警告がアラート通報されます。
  この機能を利用し、サーバが高負荷状態でのモニタリソースへの監視処理時間を算出し、モニタリソースの監視タイムアウト時間を決定することができます。

- 遅延警告割合に100を設定した場合

  遅延警告の通報を行いません。

注: テスト運用以外で、0%などの低い値を設定しないように注意してください。

関連情報: モニタリソースの遅延警告は[クラスタプロパティ]→[遅延警告]タブの[モニタ遅延警告]で設定します。

モニタリソースの監視開始待ち

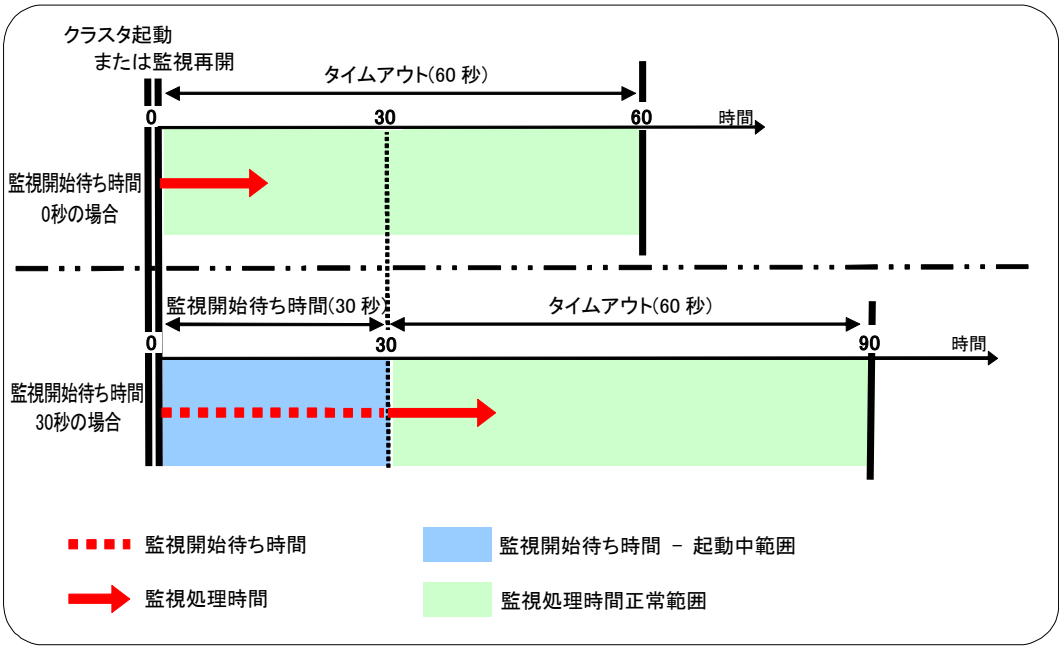
監視開始待ちとは、監視を指定した監視開始待ち時間後から開始することをいいます。

以下は、監視開始待ちを 0 秒に指定した場合と 30 秒に指定した場合の監視の違いを時系列で表した説明です。

[モニタリソース構成]

<監視>

インターバル	30 秒
タイムアウト	60 秒
リトライ回数	0 回
監視開始待ち時間	0 秒 / 30 秒



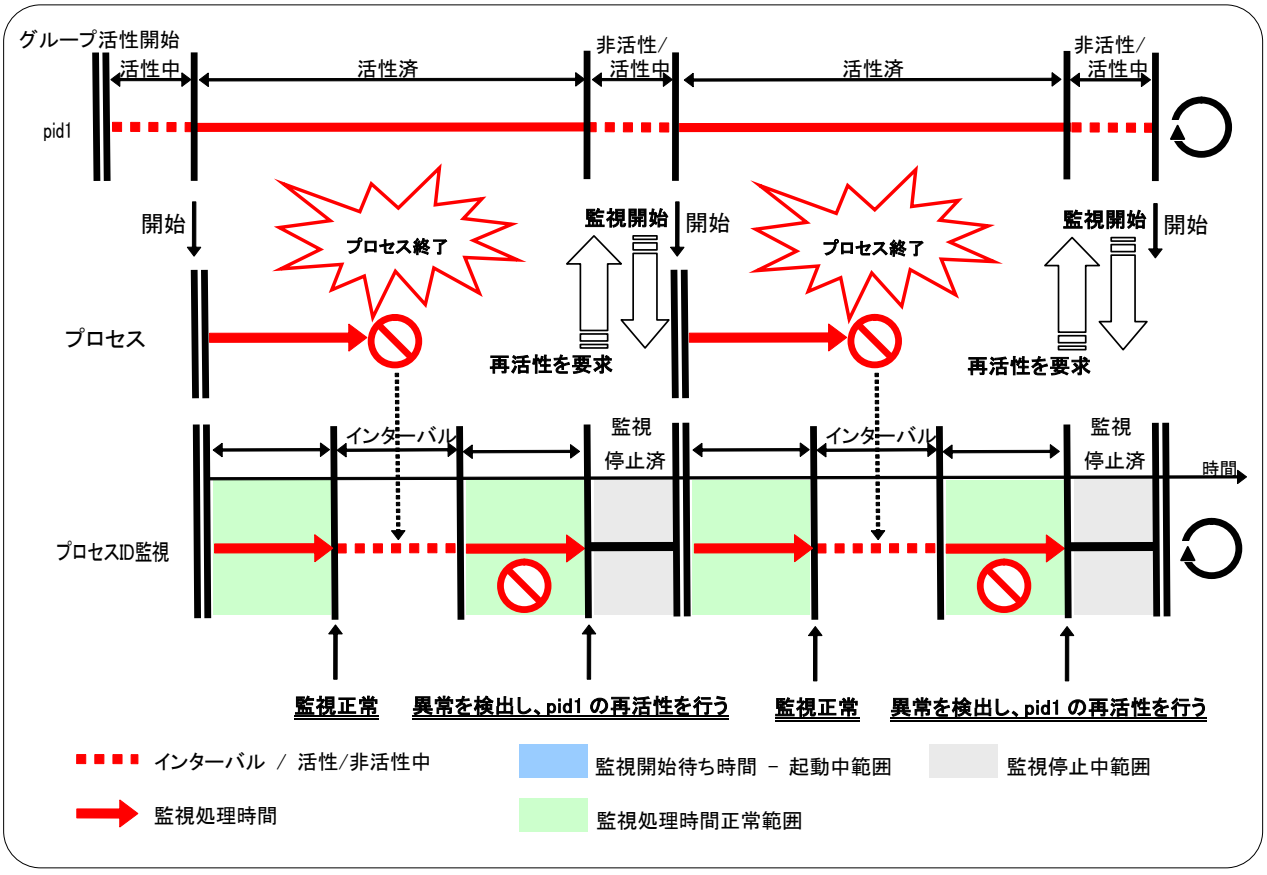
注: 監視制御コマンドによるモニタリソースの一時停止/再開を行った場合も、指定された監視開始待ち時間後に再開します。

監視開始待ち時間は、PID 監視リソースが監視する EXEC リソースのようにアプリケーションの設定ミスなどにより監視開始後すぐに終了する可能性があり、再活性化では回復できない場合に使用します。

たとえば、以下のように監視開始待ち時間を 0 に設定すると回復動作を無限に繰り返す場合があります。

[PID 監視リソース構成]

<監視>	
インターバル	5 秒
タイムアウト	60 秒
リトライ回数	0 回
監視開始待ち時間	0 秒(既定値)
<異常検出>	
回復動作	回復対象を再起動
回復対象	exec
再活性化しきい値	1 回
最終動作	グループ停止



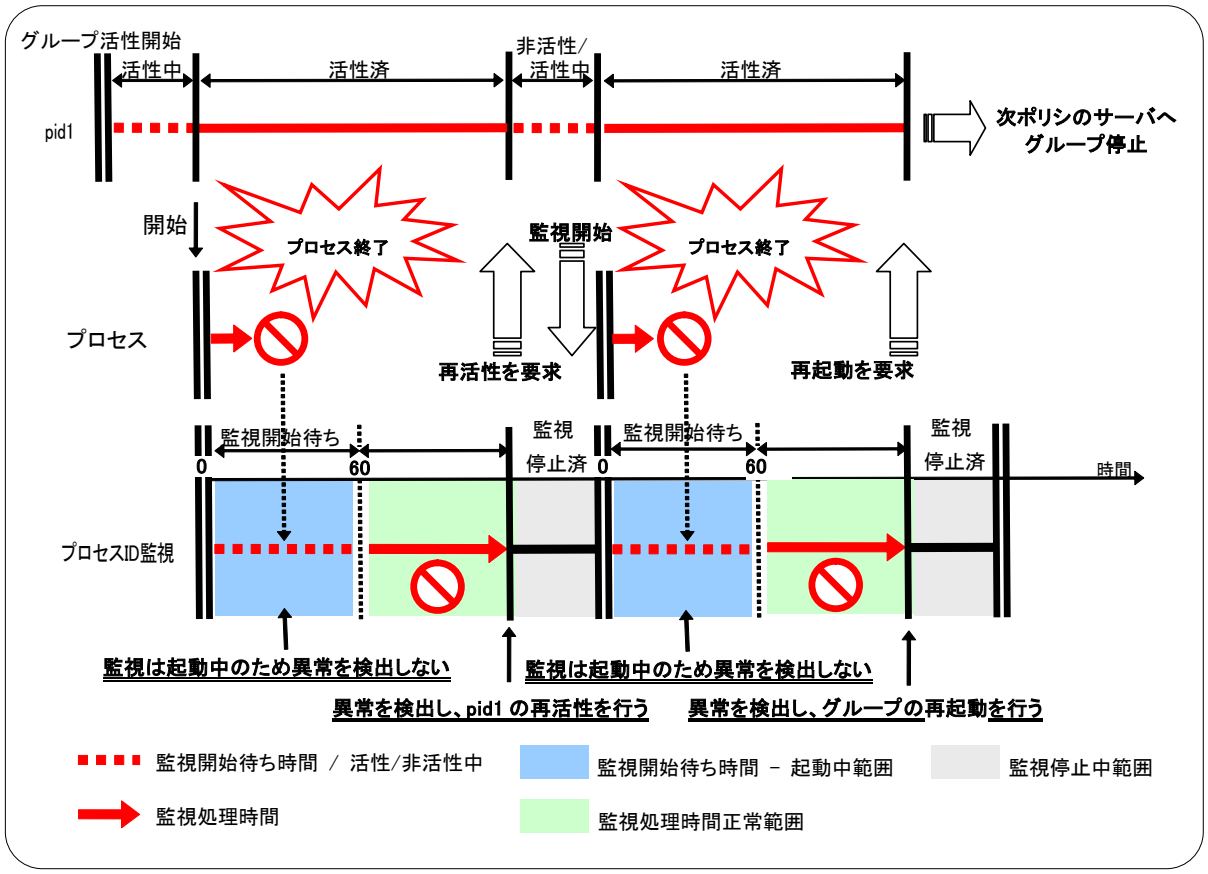
この回復動作を無限に繰り返す原因は、初回の監視処理が正常終了することにあります。モニタリソースの回復動作の現在回数は、モニタリソースが正常状態になればリセットされます。そのため、現在回数が常に 0 リセットされ再活性化の回復動作を無限に繰り返すことになります。

上記の現象は、監視開始待ち時間を設定することで回避できます。

監視開始待ち時間には、アプリケーションが起動後、終了しうる時間として既定値で 60 秒を設定しています。

[PID 監視リソース構成]

<監視>	
インターバル	5 秒
タイムアウト	60 秒
リトライ回数	0 回
監視開始待ち時間	60 秒
<異常検出>	
回復動作	回復対象を再起動
回復対象	exec
再活性化しきい値	1 回
最終動作	グループ停止



再起動回数制限について

活性異常、非活性異常検出時の最終動作、またはモニタリソース異常検出時の最終動作として OS の再起動を伴うような設定をしている場合に、シャットダウン回数、または再起動回数を制限することができます。

注: 再起動回数はサーバごとに記録されるため、最大再起動回数はサーバごとの再起動回数の上限になります。

また、グループ活性、非活性異常検出時の最終動作による再起動回数とモニタリソース異常の最終動作による再起動回数は別々に記録されます。

最大再起動回数をリセットする時間に 0 を設定した場合には、再起動回数はリセットされません。リセットする場合は clpregctrl コマンドを使用する必要があります。

セクション V　リリースノート

このセクションでは、CLUSTERPRO X SingleServerSafe の制限事項や、既知の問題とその回避策について説明します。

第 9 章　　注意制限事項

第 9 章　　注意制限事項

本章では、注意事項や既知の問題とその回避策について説明します。

本章で説明する項目は以下の通りです。

システム構成検討時

システム構成時に留意すべき事項について説明します。

Builder、WebManagerの動作OSについて

- x86_64 のマシン上で Builder および、WebManager を動作させるには 32bit 用の Web ブラウザおよび Java Runtime を使用する必要があります。

JVMモニタリソースについて

- 同時に監視可能な Java VM は最大 25 個です。同時に監視可能な Java VM とは Builder(監視(固有)タブ→識別名)で一意に識別する Java VM 数のことです。
- Java VM と Java Resource Agent 間のコネクションは SSL には対応していません。
- Java VM を監視する時、監視対象と同一の名称を持つ別のプロセスが存在する場合、異なる監視対象に対して C ヒープ監視をする可能性があります。
- スレッドのデッドロックは検出できない場合があります。これは、Java VM の既知で発生している不具合です。詳細は、Oracle の Bug Database の「Bug ID: 6380127 」を参照してください。
- WebOTX のプロセスグループを監視する時、プロセスの多重度が 2 以上になると監視を行うことはできません。WebOTX V8.4 以降では監視可能です。
- Java Resource Agent が監視できる Java VM は、JVM モニタリソースが動作中のサーバと同じサーバ内のみです。
- Java Resource Agent が監視できる JBoss のサーバインスタンスは、1 サーバに 1 つまでです。
- Builder(クラスタプロパティ→JVM 監視タブ→Java インストールパス)で設定した Java インストールパスは、クラスタ内のサーバにおいて、共通の設定となります。JVM 監視が使用する Java VM のバージョンおよびアップデートは、クラスタ内のサーバにおいて、同じものにしてください。
- Builder(クラスタプロパティ→JVM 監視タブ→接続設定ダイアログ→管理ポート番号)で設定した管理ポート番号は、クラスタ内のサーバにおいて、共通の設定となります。
- x86_64 版 OS 上において IA32 版の監視対象のアプリケーションを動作させている場合、または IA32 版 OS 上において x86_64 版の監視対象のアプリケーションを動作させている場合は、監視を行うことはできません。
- Builder(クラスタプロパティ→JVM 監視タブ→最大 Java ヒープサイズ)で設定した最大 Java ヒープサイズを 3000 など大きな値に設定すると、Java Resource Agent が起動に失敗します。システム環境に依存するため、システムのメモリ搭載量を元に決定してください。
- ロードバランサ連携の監視対象 Java VM の負荷算出機能を利用する場合は、SingleServerSafe での利用を推奨します。また、Red Hat Enterprise Linux でのみ動作可能です。
- 監視対象 Java VM の起動オプションに「-XX:+UseG1GC」が付加されている場合、Java 7 以前では JVM モニタリソースの[プロパティ]-[監視(固有)] タブ-[調整]プロパ

ティ-[メモリ]タブ内の設定項目は監視できません。
Java 8 以降では JVM モニタリソースの[プロパティ]-[監視(固有)] タブ-[JVM 種別]に[Oracle Java(usage monitoring)]を選択することで監視可能です。

メール通報について

- メール通報機能は、STARTTLSやSSLに対応していません。

構成情報作成時

構成情報の設計、作成前にシステムの構成に依存して確認、留意が必要な事項です。

環境変数

環境変数が 256 個以上設定されている環境では、下記のスクリプトが実行できません。下記の機能またはリソースを使用する場合は、環境変数を 255 個以下に設定してください。

- exec リソースが活性/非活性時に実行する開始/停止スクリプト
- カスタムモニタリソースが監視時に実行するスクリプト
- グループリソース、モニタリソース異常検出後の最終動作実行前スクリプト

サーバのリセット、パニック、パワーオフ

CLUSTERPRO X SingleServerSafe が「サーバのリセット」または「サーバのパニック」または「サーバのパワーオフ」を行う場合、サーバが正常にシャットダウンされません。そのため下記のリスクがあります。

- マウント中のファイルシステムへのダメージ
- 保存していないデータの消失

「サーバのリセット」または「サーバのパニック」が発生する設定は下記です。

- グループリソース活性時/非活性時異常時の動作
  -sysrq パニック
  -keepalive リセット
  -keepalive パニック
  -BMC リセット
  -BMC パワーオフ
  -BMC サイクル
  -BMC NMI
- モニタリソース異常検出時の最終動作
  -sysrq パニック
  -keepalive リセット
  -keepalive パニック
  -BMC リセット
  -BMC パワーオフ
  -BMC サイクル
  -BMC NMI
- ユーザ空間監視のタイムアウト検出時動作
  -監視方法 softdog
  -監視方法 ipmi
  -監視方法 keepalive

注:「サーバのパニック」は監視方法が keepalive の場合のみ設定可能です。

- シャットダウン監視
  -監視方法 softdog
  -監視方法 ipmi

-監視方法 keepalive

注:「サーバのパニック」は監視方法が keepalive の場合のみ設定可能です。

グループリソースの非活性異常時の最終アクション

非活性異常検出時の最終動作に「何もしない」を選択すると、グループが非活性失敗のまま停止しません。
本番環境では「何もしない」は設定しないように注意してください。

VxVM が使用する RAW デバイスの確認

ボリューム RAW デバイスの実 RAW デバイスについて事前に調べておいてください。

CLUSTERPRO をインストールする前に、片サーバで活性しうる全てのディスクグループをインポートし、全てのボリュームを起動した状態にします。

以下のコマンドを実行します。

```
# raw -qa
/dev/raw/raw2:  bound to major 199, minor 2
/dev/raw/raw3:  bound to major 199, minor 3
```

①　②

例)ディスクグループ名、ボリューム名がそれぞれ以下の場合

ディスクグループ名 dg1

dg1 配下のボリューム名 vol1、vol2

1. 以下のコマンドを実行します。

```
# ls -l /dev/vx/dsk/dg1/
brw-------   1 root     root     199,   2 5月 15 22:13 vol1
brw-------   1 root     root     199,   3 5月 15 22:13 vol2
```

③

2. ② と ③ のメジャー/マイナ番号が等しいことを確認します。

これにより確認された RAW デバイス ① は監視方法が READ (VxVM) 以外のディスクモニタリソースでは絶対に設定しないでください。

遅延警告割合

遅延警告割合を 0 または、100 に設定すれば以下のようなことを行うことが可能です。

- 遅延警告割合に0を設定した場合

  監視毎に遅延警告がアラート通報されます。
  この機能を利用し、サーバが高負荷状態での監視リソースへのポーリング時間を算出し、監視リソースの監視タイムアウト時間を決定することができます。

- 遅延警告割合に100を設定した場合
  遅延警告の通報を行いません。

テスト運用以外で、0%等の低い値を設定しないように注意してください。

ディスクモニタリソースの監視方法TURについて

- SCSIのTest Unit Readyコマンドや SG_IOコマンドをサポートしていないディスク、ディスクインターフェイス(HBA)では使用できません。
  ハードウェアがサポートしている場合でもドライバがサポートしていない場合があるのでドライバの仕様も合わせて確認してください。
- S-ATAインターフェイスのディスクの場合には、ディスクコントローラのタイプや使用するディストリビューションにより、OSにIDEインターフェイスのディスク(hd)として認識される場合とSCSIインターフェイスのディスク(sd)として認識される場合があります。
  IDEインターフェイスとして認識される場合には、すべてのTUR方式は使用できません。
  SCSIインターフェイスとして認識される場合には、TUR(legacy)が使用できます。
  TUR(generic)は使用できません。
- Read方式に比べてOSやディスクへの負荷は小さくなります。
- Test Unit Readyでは、実際のメディアへのI/Oエラーは検出できない場合があります。

WebManagerの画面更新間隔について

- WebManagerタブの「画面データ更新インターバル」には、基本的に30秒より小さい値を設定しないでください。

スクリプトのコメントなどで取り扱える2バイト系文字コードについて

- CLUSTERPROでは、Linux環境で編集されたスクリプトはEUC、Windows環境で編集されたスクリプトはShift-JISとして扱われます。その他の文字コードを利用した場合、環境によっては文字化けが発生する可能性があります。

統合 WebManager 用 IP アドレス(パブリック LAN IP アドレス)の設定について

- CLUSTERPRO X2.1 以前のバージョンの[パブリック LAN IP アドレス]の設定は[クラスタプロパティ] - [WebManager タブ] の[統合 WebManager 用 IP アドレス]で設定できます。

システムモニタリソースの設定について

- リソース監視の検出パターン
  System Resource Agent では、「しきい値」、「監視継続時間」という2つのパラメータを組み合わせて検出を行います。
  各システムリソース(オープンファイル数、ユーザプロセス数、スレッド数、メモリ使用量、CPU 使用率、仮想メモリ使用量)を継続して収集し、一定時間（継続時間として指定した時間）しきい値を超えていた場合に異常を検出します。

外部連携モニタリソースの設定について

- 外部連携モニタリソースに異常を通知するには、[clprexec] コマンドを用いる方法があります。
- [clprexec] コマンドを用いる場合は CLUSTERPRO CD に同梱されているファイルを利用します。通知元サーバの OS やアーキテクチャに合わせて利用してください。また、通知元サーバと通知先サーバの通信が可能である必要があります。

JVM監視の設定について

- 監視対象がWebLogic Serverの場合、JVMモニタリソースの以下の設定値については、システム環境(メモリ搭載量など)により、設定範囲の上限に制限がかかることがあります。

  [ワークマネージャのリクエストを監視する]-[リクエスト数]

  [ワークマネージャのリクエストを監視する]-[平均値]

  [スレッドプールのリクエストを監視する]-[待機リクエスト リクエスト数]

  [スレッドプールのリクエストを監視する]-[待機リクエスト 平均値]

  [スレッドプールのリクエストを監視する]-[実行リクエスト リクエスト数]

  [スレッドプールのリクエストを監視する]-[実行リクエスト 平均値]

- 監視対象のJRockit JVM が64bit 版の場合、JRockit JVMから取得した各最大メモリ量がマイナスとなり使用率が計算できないため、以下のパラメータが監視できません。

  [ヒープ使用率を監視する]- [領域全体]

  [ヒープ使用率を監視する]- [Nursery Space]

  [ヒープ使用率を監視する]- [Old Space]

  [非ヒープ使用率を監視する]- [領域全体]

  [非ヒープ使用率を監視する]- [ClassMemory]

- Java Resource Agentを使用するには、「インストールガイド」の「JVMモニタの動作環境」に記載しているJRE(Java Runtime Environment)をインストールしてください。監視対象（WebLogic ServerやWebOTX）が使用するJREと同じ物件を使用することも、別の物件を使用することも可能です。
- モニタリソース名に空白を含まないでください。
- 異常検出時に障害原因別にコマンドを実行するための[コマンド]とロードバランサ連携機能は併用できません。

CLUSTERPROの構成変更時

クラスタとして運用を開始した後に構成を変更する場合に発生する事象で留意して頂きたい事項です。

リソースプロパティの依存関係について

リソースの依存関係を変更した場合、クラスタサスペンド、リジュームにより変更が反映されます。

リソースの依存関係と反映方法としてリソース停止が必要な設定変更をした場合、リジューム後のリソースの起動状態が依存関係を考慮したものになっていない場合があります。

次回グループ起動時から正しく依存関係の制御が行われるようになります。

グループリソースの追加、削除について

同一グループリソース名を別のグループへ移す設定変更を行う場合、以下の手順にて行ってください。

以下の手順にて行わなかった場合、正常に動作できなくなる可能性があります。

例) EXEC リソース exec1 をグループ failover1 から別のグループ failover2 に移す場合

1. グループ failover1 から exec1 を削除します。
2. 設定の反映を行います。
3. exec1 をグループ failover2 へ追加します。
4. 設定の反映を行います。

登録最大数一覧

	Builder Version	登録最大数
サーバ	3.0.0-1以降	1
グループ	3.1.0-1未満	64
	3.1.0-1以降	128
グループリソース (1グループにつき)	3.1.0-1未満	128
	3.1.0-1以降	512
モニタリソース	3.0.0-1以降	512
システムモニタリソース	3.1.0-1以降	1

付録 A　　索引

W

X

あ

い

お

か

く

け

こ

さ

し

す

せ

そ