

CA Spectrum®

AlarmNotifier ユーザ ガイド

リリース 9.4



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。

CA の事前の書面による承諾を受けずに本ドキュメントの全部または一部を複製、譲渡、開示、変更、複製することはできません。本ドキュメントは、CA が知的財産権を有する機密情報です。ユーザは本ドキュメントを開示したり、
(i) 本ドキュメントが関係する CA ソフトウェアの使用について CA とユーザとの間で別途締結される契約または (ii) CA とユーザとの間で別途締結される機密保持契約により許可された目的以外に、本ドキュメントを使用することはできません。

上記にかかわらず、本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負いません。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本ドキュメントの制作者は CA です。

「制限された権利」のもとの提供: アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2014 CA. All rights reserved. 本書に記載された全ての製品名、サービス名、商号およびロゴは各社のそれぞれの商標またはサービスマークです。

CA Technologies 製品リファレンス

このドキュメントでは、CA Spectrum® について説明します。

CA への連絡先

テクニカルサポートの詳細については、弊社テクニカルサポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。

目次

第 1 章: AlarmNotifier の概要	7
AlarmNotifier について	7
アラーム監視プロセス	8
CA Spectrum Alarm Notification Manager (SANM)	9
第 2 章: AlarmNotifier の操作	11
AlarmNotifier の起動.....	11
AlarmNotifier の停止.....	12
プロセス デーモンによる AlarmNotifier の起動	13
複数の AlarmNotifier の実行.....	15
AlarmNotifier 出力.....	16
スクリプト パラメータ定義.....	17
永続的なアラームと古くなったアラーム	20
第 3 章: AlarmNotifier のカスタマイズ	23
スクリプトの変更.....	23
スクリプト内での電子メール通知の有効化.....	23
Windows プラットフォーム上のメール サービス	24
スクリプト出力の制限.....	26
CA Spectrum 属性をスクリプトに渡す	27
グローバルアラーム属性.....	29
サードパーティ アプリケーションへのデータの送信	30
引数: USE_NEW_INTERFACE を True に設定した場合.....	31
引数: USE_NEW_INTERFACE を False に設定した場合.....	32
日時	34
.alarmrc リソース ファイルのカスタマイズ.....	34
.alarmrc のパラメータ	35
CA Spectrum の再インストールまたはアップグレード	37

第 1 章: AlarmNotifier の概要

このセクションには、以下のトピックが含まれています。

[AlarmNotifier について \(P. 7\)](#)

[アラーム監視プロセス \(P. 8\)](#)

[CA Spectrum Alarm Notification Manager \(SANM\) \(P. 9\)](#)

AlarmNotifier について

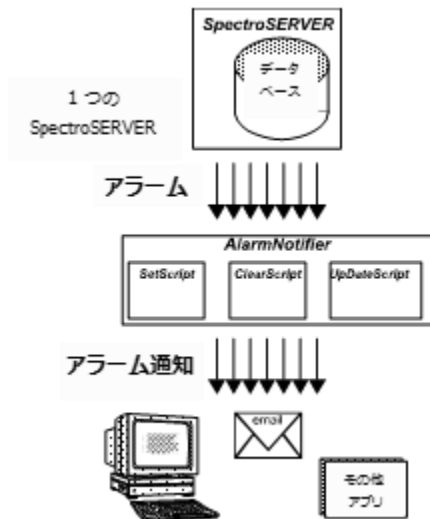
AlarmNotifier は、コア CA Spectrum コンポーネントと共にインストールされる SpectroSERVER クライアントアプリケーションです。AlarmNotifier アプリケーションは単一の SpectroSERVER に接続し、CA Spectrum アラームステータスに関する通知を提供するスクリプトを呼び出します。

端末シェル コマンドプロンプトから AlarmNotifier を起動します。起動されると、アラームが設定、クリア、または更新されるたびに呼び出されるスクリプトからの出力を継続的に表示します。AlarmNotifier は CA Spectrum に対して以下の機能を提供します。

- 単一の SpectroSERVER のアラーム監視
- アラーム情報を生成する 3 つのスクリプト：SetScript、ClearScript、および UpdateScript
これらのスクリプトには、ユーザの環境に合わせてカスタマイズできる設定が含まれます。
- AlarmNotifier の運用上の機能を変更するように設定できるリソースファイルパラメータ

アラーム監視プロセス

AlarmNotifier は、CA Spectrum のアラーム監視および通知機能を補います。以下の図は、AlarmNotifier と CA Spectrum の関係を示しています。



CA Spectrum が一部のアラーム機能を実行する一方で、AlarmNotifier が他の機能を実行します。CA Spectrum はモデリングされたネットワーク エレメントをポーリングし、SpectroSERVER データベースに格納される各エレメントに関するステータス情報を更新します。

CA Spectrum は、ネットワークからトラップを受信するとき、またはネットワーク エレメント モデルの重大ステータスの変更を検出するときに、アラームを生成します。OneClick トポロジ ビューで、モデルアイコンの状態が緑から別の色に変わることによって、アラームの重大度を示します。CA Spectrum は、[アラーム] タブ内のアラームに関する情報をポストします。アラームのイベント情報は、OneClick コンテンツ画面内の [イベント] タブに表示されます。

AlarmNotifier が起動されると、CA Spectrum に登録されます。次に、AlarmNotifier という名前の ClientApp タイプのモデルが作成されます。このモデルは CA Spectrum トポロジのどのビューでも表示できません。ただし、[イベント] タブで参照できます。[イベント] タブには、このモデルのアプリケーション起動および停止時間などの情報が表示されます。

AlarmNotifier は SpectroSERVER にクエリを実行して、既存のアラームに関する情報を要求します。AlarmNotifier はスクリプトを実行し、既存のアラームに関する通知を生成します。

アラームが設定、クリア、または更新されるたびに、AlarmNotifier は SpectroSERVER から情報を受信し、関連するスクリプトを呼び出します。AlarmNotifier スクリプトは、ネットワーク担当者に送信されるアラームの電子メール通知を開始できます。また、スクリプトによってサードパーティ アプリケーションにアラーム情報を転送できます。

CA Spectrum Alarm Notification Manager (SANM)

CA Spectrum Alarm Notification Manager (SANM) は、AlarmNotifier の機能を強化できる、CA Spectrum のアドオン コンポーネントです。SANM ポリシー管理者によって、複数のアラーム フィルタ ポリシーを作成できます。ポリシーは、AlarmNotifier アプリケーションの一意に命名されたインスタンスと関連付けることができます。これらのポリシーを使用して、関連性があると見なしたアラームについてのみ通知を生成するように AlarmNotifier に指示します。

SANM ポリシー管理者では、必要に応じて AlarmNotifier アプリケーションとポリシーを関連付けることができます。スケジューラ ユーティリティを使用して、この関連付けプロセスを自動化できます。

SANM をインストールすると、AlarmNotifier は以下の機能を提供します。

- 分散 SpectroSERVER アラーム監視
- より多くのアラーム情報を提供する追加のスクリプト パラメータ
- アラームを確認しクリアするためのコマンド
- AlarmNotifier アクティビティをログに記録し、同時に AlarmNotifier の複数インスタンスを実行できるようにする追加の起動オプション
- SANM アラーム フィルタリング ツール

注: SANM の詳細については、「Alarm Notification Manager ユーザガイド」を参照してください。

第 2 章: AlarmNotifier の操作

このセクションには、以下のトピックが含まれています。

[AlarmNotifier の起動](#) (P. 11)

[AlarmNotifier の停止](#) (P. 12)

[プロセスデーモンによる AlarmNotifier の起動](#) (P. 13)

[AlarmNotifier 出力](#) (P. 16)

[スクリプトパラメータ定義](#) (P. 17)

[永続的なアラームと古くなったアラーム](#) (P. 20)

AlarmNotifier の起動

AlarmNotifier は以下のディレクトリにインストールされます。

`<$SPECROOT>/Notifier`

デフォルトでは、このディレクトリには以下のファイルが含まれています。

- `.alarmrc`
- `AlarmNotifier`
- `ClearScript`
- `README`
- `SetScript`
- `UpdateScript`

SANM がインストールされている場合、追加ファイルはこのディレクトリの中にあります。詳細については、「Alarm Notification Manager ユーザガイド」を参照してください。

注: Windows 環境で、初めて AlarmNotifier を起動する前にメール サービスを設定します。

次の手順に従ってください:

1. <\$SPECROOT>/Notifier/.alarmrc リソース ファイル内の LANDSCAPE パラメータが、接続しようとしているランドスケープを指定していることを確認します。
2. このランドスケープが稼働中であることを確認します。
3. デフォルトのリソース ファイル (.alarmrc) または変更されたバージョンのリソース ファイルのいずれかを使用して、次のように AlarmNotifier を起動します。

- デフォルトのリソース ファイルを使用して AlarmNotifier を起動する方法:

```
<$SPECROOT>/Notifier/AlarmNotifier
```

- 別のリソース ファイルを使用して AlarmNotifier を起動する方法:

```
<$SPECROOT>/Notifier/AlarmNotifier -r <resource file>
```

注: SANM を使用している場合は、AlarmNotifier コマンドの追加のオプションを利用できます。詳細については、「Alarm Notification Manager ユーザ ガイド」を参照してください。

詳細情報:

[Windows プラットフォーム上のメール サービス \(P. 24\)](#)

AlarmNotifier の停止

AlarmNotifier を起動すると、継続的に実行されます。AlarmNotifier を停止するには、以下のいずれかのコマンドを使用します。

フォアグラウンドで実行されている AlarmNotifier の停止

次の手順に従ってください:

Windows/Solaris

以下を入力します。

Control+C

バックグラウンドで実行されている AlarmNotifier の停止

次の手順に従ってください:

Windows

AlarmNotifier プロセスを終了するには、`$SPECROOT/lib/SDPM/kill -TERM <pid>` を使用します。

Solaris

以下を入力します。

```
kill <option> <AlarmNotifier PID>
```

プロセス デーモンによる AlarmNotifier の起動

`processd` (CA Spectrum プロセス管理デーモン) を使用して AlarmNotifier プロセスの起動を自動化できます。

`processd` は、CA Spectrum インストール中およびシステムが再起動されるたびに自動的に起動されます。`processd` は起動すると、自動的に他のプロセスを起動して管理します。

`processd` で AlarmNotifier アプリケーションを起動して追跡できるようにするには、インストールチケットファイルを作成します。

注: `AlarmNotifier.idb` はデフォルトで付属しています。そのため、ゼロからファイルを作成する必要はありません。`AUTOBOOTSTART` は Y に変更できます。

次の手順に従ってください:

1. テキスト エディタを使用して、以下の内容でファイルを作成します。

```
# Processd Install Ticket for Alarm Notifier
PARTNAME;ALARMNOTIFIER;
APPNAME;Alarm Notifier;
WORKPATH;$SPECROOT/Notifier;
LOGNAMEPATH;$WORKPATH/ALARMNOTIFY.OUT;
ADMINPRIVS;y;
AUTORESTART;y;
AUTOBOOTSTART;y;
#STATEBASED;N;
NUMPROCS;1;
RETRYTIMEOUT;6000;
TICKETUSER;<USERNAME>;
RETRYMAX;20;
STARTPRIORITY;30;
#ENV;<var>=<value>;
ARGV;$WORKPATH/AlarmNotifier<CSEXE>; //
```

2. このファイルを AlarmNotifier.idb という名前にして、以下のディレクトリに保存します。

```
<$SPECROOT>/lib/SDPM/partslst
```

LOGNAMEPATH パラメータは、AlarmNotifier アプリケーションのログファイルの名前およびパスを指定します。

AlarmNotifier が起動するたびに、新しいログファイルが生成され、前のログファイルのバックアップが作成されます。ただし、SpectroSERVER は最新の 2 つの AlarmNotifier ログファイルのみを格納します。AlarmNotifier のログファイル出力をすべて保管するには、ログファイル出力を個別のファイルに保存するスクリプトを作成します。

TICKETUSER パラメータの値 <USERNAME> は、有効な CA Spectrum ユーザモデルのユーザ名である必要があります。

AlarmNotifier を起動する前に、SpectroSERVER が実行されている必要があります。そのため、STARTPRIORITY パラメータを 30 に設定して、AlarmNotifier が SpectroSERVER に依存することを指定してください。詳細については、「分散 SpectroSERVER 管理者ガイド」を参照してください。

複数の AlarmNotifier の実行

CA Spectrum processd の起動時に、異なる通知ポリシーが設定された複数の AlarmNotifier を起動できます。このような AlarmNotifier に必要なインストールチケットファイルを作成することで、このセットアップを有効にできます。複数の AlarmNotifier を実行するには、以下のタスクを実行します。

- 複数の AlarmNotifier に必要なインストール チケット ファイルを作成するか、デフォルトの ALARMNOTIFIER.idb ファイルのコピーを作成してファイルの名前を適宜変更します。詳細については、「[プロセス デーモンによる AlarmNotifier の起動 \(P. 13\)](#)」を参照してください。
- インストール チケット ファイルが作成されたら、インストール チケット ファイルの値を変更して、目的の通知ポリシーを設定した AlarmNotifier を起動できます。以下の値を指定できます。

APPNAME

通知ポリシーを実装するために作成されるアプリケーションの名前を指定します。詳細については、「Alarm Notification Manager ユーザ ガイド」を参照してください。

AUTOBOOTSTART

processd の起動時に、プロセスを起動するようプロセス デーモンに通知します。値を Y に設定します。

WORKPATH

アプリケーションが作成される作業パスを指定します。

LOGNAMEPATH

AlarmNotifier アプリケーションのログ ファイルの名前およびパスを指定します。

注: LOGNAMEPATH は、各インスタンスに固有のものにする必要があります。

ARGV

次の値を指定します。

```
$SPECROOT/Notifier/AlarmNotifier<CSEXE> -r alarmrc -n <APPNAME に指定したアプリケーションの名前>
```

注: 別のインスタンスとして実行するために AlarmNotifier 実行可能ファイルを別のディレクトリにコピーする場合は、WORKPATH、LOGNAMEPATH、および ARGV をそれに応じてカスタマイズします。

AlarmNotifier 出力

AlarmNotifier は、CA Spectrum 内でアラームが設定、クリア、または更新されたことを検出するたびに、SetScript、ClearScript、または UpdateScript を呼び出します。各スクリプトはアラームのステータス（設定、クリア、または更新）に関する情報を含む通知を生成し、それを表示します。通知にはそれぞれ、「[スクリプトパラメータ定義 \(P. 17\)](#)」で定義されているパラメータが含まれます。

SetScript

以下の状況のアラームに対して呼び出されます。

- AlarmNotifier が起動され、既存のアラームを検出します。alarmrc リソース ファイルで GET_EXISTING_ALARMS の値が 'false' に設定されていない場合、AlarmNotifier は SetScript を呼び出します。

デフォルト : true

- AlarmNotifier が実行されている間、CA Spectrum はアラームを生成します。

ClearScript

アラームがクリアされる時に呼び出されます。

UpdateScript

アラームが更新される時に呼び出されます。アラームは、以下の状況で更新されるように定義されます。

- トラブルシュータがアラームに割り当てられました。または、トラブルシュータ名が変更されました。スクリプト内の RepairPerson パラメータはこのトラブルシュータ名を表します。
- アラームのステータスが変わりました。アラームのステータス情報は [アラーム] タブに入力されます。AlarmStatus パラメータはスクリプトでステータスを表します。
- アラームは CA Spectrum で確認または確認解除されます。
- 新規イベント、または既存のイベントへの変更は、アラーム状態であるデバイス上で発生します。

スクリプトパラメータ定義

AlarmNotifier スクリプトのパラメータを更新して、その機能をカスタマイズできます。以下のリストでは、**SetScript**、**ClearScript**、および **UpdateScript** で利用可能なパラメータについて説明します。

注: ユーザが **AlarmNotifier** を持った **SANM** を使用するとき、追加のスクリプトパラメータは利用可能です。詳細については、「**Alarm Notification Manager ユーザガイド**」を参照してください。

Date

AlarmNotifier がアラームの設定、更新、またはクリアを検出する日付を指定します。

形式: mm/dd/yyyy

Time

AlarmNotifier がアラームの設定、更新、またはクリアを検出する時刻を指定します。

形式: hh:mm:ss

Mtype

アラームが設定、更新、またはクリアされるモデルのタイプを指定します。

ModelName

アラームが設定、更新、またはクリアされるモデルの名前を指定します。**ModelName** に特殊文字が含まれる場合は、エラーを回避するために特殊文字を環境変数としてスクリプトに渡します。特殊文字は、「\$」や「*」など、コマンドシェルが特別な意味があると解釈する文字です。

モデル名を環境変数として渡すには、**.alarmrc** ファイル内の **EXTRA_ATTRS_AS_ENVVARS** パラメータの値に **ModelName** の属性 ID (0x1006e) を追加します。詳細については、「[CA Spectrum 属性をスクリプトに渡す \(P. 27\)](#)」および「[.alarmrc のパラメータ \(P. 35\)](#)」を参照してください。

AlarmID

CA Spectrum がアラームに割り当てた数字の識別子を指定します。

Global AlarmID

CA Spectrum がアラームに割り当てる一意の数字の識別子を指定します。AlarmID とは異なり、グローバルアラーム ID は、CA Spectrum 環境内で一意であるだけでなく、一意の識別子として他の環境に渡すこともできます。一意の識別子をサードパーティ ソフトウェアへ渡すには、この値を使用します。デフォルトでは、Global AlarmID は各 AlarmNotifier スクリプトでコメントアウトされています。このパラメータを渡すには、コメント記号 (#) を削除します。

Severity

アラームの CA Spectrum 重大度レベルコードを指定します。重大 (赤)、メジャー (オレンジ)、マイナー (黄色)、保守 (茶色)、抑制 (灰色)、または初期 (青)。

ProbableCauseID

アラーム用の想定される原因と関連付けられているの 16 進数の識別子を指定します。

RepairPerson

[アラーム] タブでアラームに割り当てられるトラブルシュータを指定します。トラブルシュータが最初に割り当てられるとき、およびそれ以降に割り当てられるたびに、AlarmNotifier は UpdateScript を呼び出します。

トラブルシュータ (修復担当者) の名前が SetScript と ClearScript が生成する通知に表示されるかどうかは、以下の状況によって決まります。

- アラームが設定されたことを AlarmNotifier が検出した後でトラブルシュータが割り当てられる場合、SetScript は名前を表示しません。UpdateScript と ClearScript は名前を表示します。
- AlarmNotifier が (AlarmNotifier が起動される前に存在していたアラームの) セットを検出する前にトラブルシュータが割り当てられる場合、3 つのスクリプトはすべてトラブルシュータの名前を表示します。

注: 詳細については、「オペレータ ガイド」を参照してください。

AlarmStatus

OneClick 内のアラーム用のステータス情報を示します。ステータス情報が最初に入力される時、およびそれ以降に入力されるたびに、AlarmNotifier は UpdateScript を呼び出します。ステータス情報は、通常 SetScript と ClearScript が生成する通知に表示されます。以下の状況は例外です。

- アラームが設定されたことを AlarmNotifier が検出した後でステータス情報が入力される場合、SetScript は情報を表示しません。UpdateScript と ClearScript は情報を表示します。
- AlarmNotifier が（AlarmNotifier が起動される前に存在していたアラームの）セットを検出する前にステータス情報が入力される場合、3 つのスクリプトはすべてステータス情報を表示します。

注: 詳細については、「オペレータ ガイド」を参照してください。

SpectroSERVER

アラームが設定、更新、またはクリアされた SpectroSERVER のホストの名前を指定します。

Landscape

アラームが設定、更新、またはクリアされたランドスケープのハンドルを指定します。

ModelHandle

アラームが生成されたモデルのハンドルを指定します。

ModelTypeHandle

アラームが設定、更新、またはクリアされたモデルタイプのハンドルを指定します。

IPAddress

アラームが設定、更新、またはクリアされたネットワーク エLEMENT の IP アドレスを指定します。

SecurityString

アラームが設定、更新、またはクリアされたモデルのセキュリティ文字列を指定します。

AlarmState

アラーム状態が「既存」または「新規」のいずれであるかを指定します。

AlarmNotifier が起動される前にアラームが設定されている場合、アラームの状態は「既存」です。 .alarmrc リソースファイル内の GET_EXISTING_ALARMS パラメータが true に設定される場合、AlarmNotifier は既存のアラームの SetScript を呼び出します。

AlarmNotifier が起動された後でアラームが生成される場合、アラームの状態は「新規」です。 また、AlarmNotifier の SpectroSERVER が、以前に接続されていた、アラームが発生したランドスケープへの接続を復元した場合も、アラーム状態は「新規」です。

Acknowledged

アラームが確認されたかどうかを指定します。

UserClearable

ユーザがアラームをクリアできるかどうかを指定します。

DeviceType

アラームが設定、更新、またはクリアされたモデルの DeviceType 属性の値を指定します。 詳細については、「認定ユーザガイド」を参照してください。

Raw Alarm Time

アラームの日時は \$DATE および \$TIME として利用可能です。 形式が指定されていないアラーム時刻を取得するには、\$RAW_ALARM_TIME を参照できます。

注: 形式が指定されていないアラーム時刻は、1970年1月1日、深夜0時 UTC 以降の経過秒数で表されます。

永続的なアラームと古くなったアラーム

SpectroSERVER が停止して再起動すると、すでに存在していたアラームは存在し続けます。 これらのアラームが「永続的」アラームです。 永続的なアラーム機能によって、SpectroSERVER がシャットダウンするときに、CA Spectrum で、トラブルシュータの割り当てやステータスなどのアラーム関連の情報を保持できます。

状況によっては、SpectroSERVER がシャットダウンしてから再起動するまでの間に、アラームの根本的な原因が解決されます。アラームは、まだ OneClick 内の [アラーム] リストに表示されますが、古くなっていると考えられます。古くなったアラーム（「残余のアラーム」とも呼ばれる）はすべて手動でクリアできます。

ただし、古くなったアラーム情報は SpectroSERVER によって AlarmNotifier に転送されません。代わりに、古くなっているアラームや残余のアラームがランドスケープに存在することを示す単一の新しいアラームは生成され、AlarmNotifier に送信されます。ユーザーが手動で古くなったアラームをクリアすると、そのアラームは AlarmNotifier でもクリアされます。最後の古くなったアラームをクリアすると、古くなったアラーム通知について「クリア」が実行されます。

第 3 章: AlarmNotifier のカスタマイズ

このセクションには、以下のトピックが含まれています。

[スクリプトの変更 \(P. 23\)](#)

[スクリプト出力の制限 \(P. 26\)](#)

[CA Spectrum 属性をスクリプトに渡す \(P. 27\)](#)

[サードパーティアプリケーションへのデータの送信 \(P. 30\)](#)

[.alarmrc リソース ファイルのカスタマイズ \(P. 34\)](#)

[CA Spectrum の再インストールまたはアップグレード \(P. 37\)](#)

スクリプトの変更

AlarmNotifier スクリプトを変更して、AlarmNotifier のアクションおよび出力をカスタマイズできます。指定された受信者への電子メール通知を開始するようにスクリプトを設定できます。また、スクリプトをカスタマイズして、アラーム移行通知が提供する情報の範囲を制限したり、サードパーティアプリケーションと統合したりすることができます。

スクリプト内での電子メール通知の有効化

各 AlarmNotifier スクリプトには、アラームのトラブルシュータへの電子メール通知を有効にするために設定できる 2 つのパラメータ (SENDMAIL と VARFORMAIL) が含まれます。1 つ以上のスクリプトについて電子メール通知を有効にできます。

注: 誤って削除した場合に備えて、デフォルトのスクリプト設定を保持するために、編集する予定のデフォルト スクリプトのバックアップ コピーを作成してください。

次の手順に従ってください:

1. <\$SPECROOT>/Notifier ディレクトリ、またはスクリプトが保存されるディレクトリに移動します。
2. スクリプトをテキストエディタで開きます。
注: スクリプトはすべて連続的に実行されます。そのため、**AlarmNotifier** を停止せずに、スクリプトを編集できます。
3. スクリプトで **SENDMAIL** パラメータを **True** に設定します。
4. **VARFORMAIL** パラメータを **RepairPerson** に設定します。
5. スクリプトを保存して閉じます。

電子メールは、[アラーム] タブ内のアラームに割り当てられたトラブルシュータに送信されます。この担当者は、アラームが発生したランドスケープの承認されたユーザ（またはユーザモデル）である必要があります。

注: アラームが発生した後で、**RepairPerson** の値が設定されます。その結果、メールは設定アクションに応じて (**SetScript** を使用して) 送信できません。ただし、**SANM** も使用している場合は、設定アクションの結果として送信されるようにメールを設定できます。詳細については、「**Alarm Notification Manager ユーザガイド**」を参照してください。

Windows プラットフォーム上のメール サービス

Windows で、**AlarmNotifier** スクリプトが電子メール通知を送信できるように、メール サービスを有効にするためにメール コマンドを使用します。

AlarmNotifier が起動する前に、この手順を実行します。そうしなかった場合、メール サービスを設定してから **AlarmNotifier** を起動するように求めるエラー メッセージが表示されます。

メール コマンド パラメータ

ターミナルのウィンドウからのメール コマンドを実行するとき、いくつかのパラメータは必須です。このようなパラメータを以下に示します。

-m

リターン ホスト名です。リターン ホストは、受信メールが受信されるコンピュータです。

-h

Simple Mail Transfer Protocol (SMTP) ホスト、つまり送信電子メールが処理されるコンピュータです。

-u

ユーザ名です。

注: AlarmNotifier はポケベルに通知を送信できます。この設定をテストするには、まず有効なローカル メール アカウントに通知を送信するように AlarmNotifier を設定することをお勧めします。その後、ポケベルに通知を送信するためにメール サービスを再設定できます。

メール サービスの設定

AlarmNotifier が電子メール通知を送信できるように、Windows 上でメール サービスをセットアップできます。

次の手順に従ってください:

1. メール サーバ管理者に問い合わせ、メール コマンド パラメータの正しい値を確認します。
2. `bash` シェルを開きます。
3. 以下のコマンドを入力します。

```
mail -m your-company.com -h smtp.your-company.com -u username
```

コマンドが正常に完了すると、コマンド使用状況リストが表示されます。

4. 設定が完全であることを確認するには、`HKEY_LOCAL_USER/Software/SMail` のレジストリ エントリを表示します。
`hostname`、`smtphost`、および `username` キーに、コマンド文字列で指定した情報が含まれています。

詳細情報:

[メール コマンド パラメータ \(P. 25\)](#)

スクリプト出力の制限

AlarmNotifier スクリプトのパラメータをコメントアウトして、スクリプトからの出力の量を低減できます。

注: スクリプトを変更する前に、「[スクリプトパラメータ定義 \(P. 17\)](#)」でパラメータの説明を確認してください。抑制している情報について十分に理解しておくことをお勧めします。

次の手順に従ってください:

1. 設定する **AlarmNotifier** アプリケーションのインスタンスを閉じます。
2. デフォルトのスクリプトディレクトリ、`<$SPECROOT>/Notifier`、または編集するスクリプトが保存されているディレクトリに移動します。
3. スクリプトをテキストエディタで開きます。
4. 抑制する `echo` コマンド行の先頭にシャープ記号 (`#`) を入力することによってコメントアウトします。以下の例では、**UserClearable** パラメータがコメントアウトされます。

```
echo "SecurityString:      " $SECSTR
echo "AlarmState:        " $ALARMSTATE
echo "Acknowledged:      " $ACKD
#echo "UserClearable:     " $CLEARABLE
```

このスクリプトが通知を生成するとき、コメントアウトされたパラメータからの情報は画面に表示されません。

5. スクリプトを保存して閉じます。

注: 変数自体の代入または `shift` コマンドをコメントアウトまたは変更しないでください。これらの行を変更すると、スクリプトはアラーム情報を正しく表示しません。

CA Spectrum 属性をスクリプトに渡す

モデルの属性をアラームと共に **AlarmNotifier** に渡すことができます。モデル属性は **SetScript**、**ClearScript**、または **UpdateScript** でパラメータとして使用できます。属性を渡すには、**.alarmrc** のパラメータ **EXTRA_ATTRS_AS_ENVVARS** または **EXTRA_ATTRS_AS_ARGS** を使用します。これらのパラメータを有効にするには、**.alarmrc** の **USE_NEW_INTERFACE** パラメータを **TRUE** に設定します。

EXTRA_ATTRS_AS_ENVVARS は環境変数として属性を **AlarmNotifier** に渡します。**EXTRA_ATTRS_AS_ARGS** はコマンドライン引数として属性を渡します。ほとんどの属性については、これらのメカニズムのいずれかを使用できます。ただし、追加データを解析するスクリプトで、新しい行または特殊文字が問題を引き起こす場合は、**EXTRA_ATTRS_AS_ENVVARS** が必要です。**USE_NEW_INTERFACE=TRUE** の場合、環境変数メカニズムを使用して、**\$STATUS**、**\$EVENTMSG**、および **\$PCAUSE** を渡すことによってこの問題を回避します。

.alarmrc ファイルで、渡す **CA Spectrum** 属性 ID と等しい適切なパラメータを設定します。属性 ID は、16 進表記または 10 進表記のいずれかで参照できます。

EXTRA_ATTRS_AS_ENVVARS を使用して、環境変数として属性を渡す場合、以下の構文を使用して、スクリプト内でこの変数を参照します。

```
$SANM_<attribute_ID>
```

```
<attribute_ID>
```

参照している属性の属性 ID を指定します。**.alarmrc** ファイル内で 16 進表記を使用してこの属性を呼び出している場合、スクリプトでも 16 進表記を使用する必要があります。**.alarmrc** ファイル内で 10 進表記を使用してこの属性を呼び出している場合、スクリプトでは 10 進表記を使用します。

Windows は、自動的に環境変数を大文字に設定します。そのため、これらの変数を参照するときには、**\$SANM_0X100C5** など大文字形式を使用します。

EXTRA_ATTRS_AS_ARGS を使用して引数として属性を渡す場合、スクリプト内で変数に値を代入することによって、スクリプトでこの変数を参照できます。

```
<variable>=${<x>}
```

```
<variable>
```

属性の値を保持する変数を指定します。

```
<x>
```

渡した引数の順序と数に応じて、適切な変数番号を指定します。

例

以下の例では、.alarmrc ファイルで、AlarmNotifier に渡される 4 つのサンプル CA Spectrum 属性を示します。その後、属性はスクリプトで参照されます。

.alarmrc ファイルでの参照

```
USE_NEW_INTERFACE=TRUE
EXTRA_ATTRS_AS_ENVVARS=0x100c5,0x11f84
EXTRA_ATTRS_AS_ARGS=0x110df,0x117dc
```

スクリプトでの参照

```
#These lines read 0x110df and 0x117dc into the variables MAC_ADDRESS
#and FIRMWARE_VERSION respectively.
shift 9
MAC_ADDRESS=$1
FIRMWARE_VERSION=$2

#These lines print out the value of each attribute.(Solaris Platform)
echo "The value of attribute 0x100c5 is: " $SANM_0x100c5
echo "The value of attribute 0x11f84 is: " $SANM_0x11f84
echo "The value of attribute 0x110df is: " $MAC_ADDRESS
echo "The value of attribute 0x117dc is: " $FIRMWARE_VERSION

#These lines print out the value of each attribute.(Windows Platform)
#references to environmental variables are in uppercase
echo "The value of attribute 0x100c5 is: " $SANM_0X100C5
echo "The value of attribute 0x11f84 is: " $SANM_0X11F84
echo "The value of attribute 0x110df is: " $MAC_ADDRESS
echo "The value of attribute 0x117dc is: " $FIRMWARE_VERSION
```

詳細情報:

[.alarmrc のパラメータ \(P. 35\)](#)

グローバル アラーム属性

ここでは、CA Spectrum グローバルアラーム属性および対応する属性 ID のリストを示します。これらの属性を AlarmNotifier に渡すには、「[CA Spectrum 属性をスクリプトに渡す \(P. 27\)](#)」で説明されている方法を使用します。

Acknowledged

0x11f4d

Alarm_Source

0x11fc4

Alarm_Status

0x11f4f

Cause_Code

0x11f50

Cleared_By_User_Name

0x11f51

Creation_Date

0x11f4e

ImpactScope

0x1290e

ImpactSeverity

0x1290d

Last_Occurrence_Date

0x1321a

Occurrences

0x11fc5

Originating_Event

0x1296e

Persistent

0x12942

Primary_Alarm

0x11f54

Severity

0x11f55

Trouble_Shooter_Email

0x12a6c

Trouble_Shooter_mh

0x11fc6

Trouble_Ticket_ID

0x12022

TroubleShooter

0x11f57

User_Clearable

0x11f9b

Customer_Impact

0x12bf6

Service_Impact

0x12bf7

注: **Service_Impact** は、必ずしもサービスの現在のヘルスを反映するとは限りません。**Service_Impact** は、デバイスアラームが生成された時点でのサービスのヘルス状態を表します。サービスのヘルスのみが変わっても、**Service_Impact** は影響を受けません。

サードパーティアプリケーションへのデータの送信

SetScript、**ClearScript**、**UpdateScript** をカスタマイズまたは置換することによって、サードパーティアプリケーションとの統合を実現できます。

SetScript、ClearScript、UpdateScript の機能を補足および拡張できます。また、これらのスクリプトに CA SpectrumCLI コマンドを含めて、SpectroSERVER からより多くの情報を取得することもできます。サードパーティアプリケーションにデータを送信する独自のコードをスクリプトに追加することもできます。

既存のスクリプトの機能を使用しない場合、独自のスクリプトまたは実行可能ファイルを実行するように AlarmNotifier に指示できます。AlarmNotifier リソースファイル (.alarmrc) では、Set_Script パラメータによって、アラームが設定されたときに実行するスクリプトを制御できます。Clear_Script パラメータは、アラームがクリアされたときに実行するスクリプトを制御し、Update_Script パラメータは、アラームが更新されたとき実行するスクリプトを制御します。

デフォルトでは、Set_Script パラメータの値は SetScript、Clear_Script パラメータの値は ClearScript、Update_Script パラメータの値は UpdateScript です。これらのパラメータの値を変更して別のスクリプトを起動できるのは、アラームが設定、クリア、または更新されたときです。

カスタムスクリプトパラメータは、CA Spectrum から抽出するべきデータ、およびデータが送信されるサードパーティアプリケーションに依存します。独自のスクリプトまたは実行可能ファイルを作成するには、まず、CA Spectrum から受信側のスクリプトまたは実行可能ファイルに渡される引数を理解する必要があります。スクリプトや実行可能ファイルでは、すべての引数を CA Spectrum から正しい順序で受信する必要があります。

引数: USE_NEW_INTERFACE を True に設定した場合

以下の表では、.alarmrc の USE_NEW_INTERFACE パラメータが TRUE に設定されているときに、各スクリプトに渡される各引数の番号、名前、および形式について説明します。

USE_NEW_INTERFACE が TRUE に設定されている場合、Status 引数は環境変数として送信されます。そのため、引数の順序が影響を受けます。

引数	名前	形式
1	Date	mm/dd/yyyy
2	Time	hh:mm:ss
3	Model Type	テキスト

引数	名前	形式
4	Model Name	テキスト
5	Alarm ID	整数
6	Severity	テキスト
7	Cause	テキスト
8	Repair Person	テキスト
9	Server	テキスト
10	Landscape	16 進数
11	Model Handle	16 進数
12	Model Type Handle	16 進数
13	IP Address	xxx.xxx.xxx.xxx
14	Security String	テキスト
15	Alarm State	テキスト
16	Acknowledged	テキスト
17	Clearable	テキスト
18	Device Type	テキスト

引数: USE_NEW_INTERFACE を False に設定した場合

以下の表では、.alarmrc の USE_NEW_INTERFACE パラメータが FALSE に設定されているときに、各スクリプトに渡される各引数の番号、名前、および形式について説明します。

注: SANM に対応した AlarmNotifier で作業している場合、追加の引数が渡されます。詳細については、「Alarm Notification Manager ユーザガイド」を参照してください。

引数	名前	形式
1	Date	mm/dd/yyyy
2	Time	hh:mm:ss
3	Model Type	テキスト

引数	名前	形式
4	Model Name	テキスト
5	Alarm ID	整数
6	Severity	テキスト
7	Cause	テキスト
8	Repair Person	テキスト
9	Status	テキスト
10	Server	テキスト
11	Landscape	16 進数
12	Model Handle	16 進数
13	Model Type Handle	16 進数
14	IP Address	xxx.xxx.xxx.xxx
15	Security String	テキスト
16	Alarm State	テキスト
17	Acknowledged	テキスト
18	Clearable	テキスト
19	Device Type	テキスト

日時

以下の条件が、Set、Clear、および Update スクリプトの Date および Time 引数に適用されます。

- **SET の場合**： Date および Time は CREATION_DATE (0x11f4e) 属性に由来します。SpectroSERVER は、アラームが作成されるときにこの属性を設定します。
- **CLEAR の場合**： Date および Time は CLEAR_DATE (0x129af) 属性に由来します。SpectroSERVER は、アラームがクリアされるときにこの属性を設定します。
- **UPDATE の場合**： Date および Time は、アラームが更新されたという通知を AlarmNotifier が受信したときを反映します。SpectroSERVER はこの値を設定しません。更新が発生した正確な時間を決定するために、この値に依存しないでください。

.alarmrc リソース ファイルのカスタマイズ

.alarmrc リソース ファイル (Notifier ディレクトリに保存される) には、AlarmNotifier の運用上のパラメータが含まれます。以下の方法でリソースファイルを変更できます。

- AlarmNotifier が、AlarmNotifier が起動する前に CA Spectrum が生成したアラームを処理するかどうかを指定します。
- AlarmNotifier に渡すオプションの CA Spectrum 属性を指定します。
- SetScript、ClearScript、または UpdateScript をカスタム スクリプトに置換します。
- AlarmNotifier が接続する SpectroSERVER を指定します。
- 展開する予定がない AlarmNotifier アクションを指定するパラメータを無効にして、ネットワーク トラフィックを縮小します。

注: パラメーターをコメントアウトしても、パラメータは無効になりません。代わりに、パラメータのデフォルト値が使用されます。

次の手順に従ってください:

1. <\$SPECROOT>/Notifier ディレクトリに移動し、.alarmrc ファイルのバックアップ コピーを作成します。
2. 好みのテキスト エディタでファイルを開きます。
3. オプションのパラメータをオフにすることにより、または新しいパラメータ値の入力することによりファイルを編集します。パラメータに **False** の値を指定するか、または値を空白のままにすることによりパラメータを無効にできます。

注: 必須のパラメータを無効にしたり、パラメータを削除したりしないでください。

4. ファイルを保存して閉じ、**AlarmNotifier** を再起動します。
変更内容は、**AlarmNotifier** が再起動されるときに有効になります。

.alarmrc のパラメータ

以下のリストでは、**AlarmNotifier** で提供されるリソース ファイルパラメータについて説明します。詳細については、「**Alarm Notification Manager ユーザガイド**」を参照してください。

LANDSCAPE

AlarmNotifier が接続する初期 SpectroSERVER ホストを識別します。ここには名前を 1 つだけ入力します。LANDSCAPE が定義されていない場合、**AlarmNotifier** はデフォルトで VNM ランドスケープ マップで最初のランドスケープを使用します。情報ウィンドウにデフォルトのランドスケープ ハンドルが表示されます。

VNM_MAIL_TIMEOUT

メール サービスが SpectroSERVER からの応答を待機し、要求がキャンセルされるまでの最小時間を指定します。

デフォルト : 60,000 ミリ秒 (1 分)

VNM_CONNECT_TIME_LIMIT

AlarmNotifier と SpectroSERVER の間での初期 TCP 要求がタイムアウトするまでの最小遅延を指定します。

デフォルト : 60,000 ミリ秒 (1 分)

KEEP_ALIVE_TIMEOUT

キープアライブ要求がタイムアウトになるまでの時間の量を指定します。

デフォルト： 30,000 ミリ秒 (30 秒)

KEEP_ALIVE_INTERVAL

SpectroSERVER に送信されるキープアライブ要求間の時間の量を指定します。キープアライブ要求は、SpectroSERVER がまだ AlarmNotifier に接続されているかどうかを確認します。AlarmNotifier は、要求に対する応答を受信しない場合、SpectroSERVER から切断します。

SpectroSERVER がこれらの要求に応答するのに低速である場合、この値を大きくすることによって、AlarmNotifier が SpectroSERVER から切断されるのを防ぐことができます。

デフォルト： 60,000 ミリ秒 (1 分)

SEND_ALARM_DELAY

連続するアラーム通知間の最小の遅延を指定します。

デフォルト： 1,000 ミリ秒 (1 秒)

GET_GRAY_INITIAL_ALARMS

灰色の初期アラームを受信するかどうかを指定します。灰色または初期アラームが必要ではない場合は、このパラメータを **FALSE** に設定します。この設定は、AlarmNotifier が生成するネットワークトラフィックを縮小し、そのパフォーマンスを改善します。

GET_EXISTING_ALARMS

AlarmNotifier が呼び出されたときに存在するアラームに関するレポートを受信するかどうかを指定します。指定しない場合、AlarmNotifier が呼び出された後で発生したアラームのレポートのみを受信します。アラーム通知とアラーム通知の間には、「SpectroSERVER からのアラームをさらに待機する」という内容のメッセージが表示されます。

CA Spectrum の再インストールまたはアップグレード

CA Spectrum を再インストールするか、CA Spectrum のバージョンをアップグレードするときに、インストールプロセスは自動的に **SetScript**、**UpdateScript**、および **ClearScript** をバックアップディレクトリに保存します。デフォルトスクリプトの別名（たとえば **SetScript_version1** や **UpdateScript_modified** など）で保存したバージョンは、`<$$SPECROOT>/Notifier` ディレクトリに保持されます。そのディレクトリには、再インストールまたはアップグレードに含まれているデフォルトスクリプトも含まれます。

さらに、インストールプロセスは、ユーザの **.alarmrc** ファイルをバックアップディレクトリに保存します。**.alarmrc** リソースファイルの別名（たとえば **.alarmrc1** または **.alarmrc2**）で保存したバージョンは、`<$$SPECROOT>/Notifier` ディレクトリで保持されます。

バックアップスクリプトおよびバックアップ **.alarmrc** は以下のディレクトリに保存されます。

```
<$$SPECROOT>/Install-Tools/SAVES_<date>/<time>/Notifier
```