

CA Spectrum®

AlarmNotifier 用户指南

版本 9.3



本文档包括内嵌帮助系统和以电子形式分发的材料（以下简称“文档”），其仅供参考，CA 随时可对其进行更改或撤销。

未经 CA 事先书面同意，不得擅自复制、转让、翻印、透露、修改或转录本文档的全部或部分内容。本文档属于 CA 的机密和专有信息，不得擅自透露，或除以下协议中所允许的用途，不得用于其他任何用途：(i) 您与 CA 之间关于使用与本文档相关的 CA 软件的单独协议；或者 (ii) 您与 CA 之间单独的保密协议。

尽管有上述规定，但如果您为本文档中所指的软件产品的授权用户，则您可打印或提供合理数量的本文档副本，供您及您的雇员内部用于与该软件相关的用途，前提是所有 CA 版权声明和标识必须附在每一份副本上。

打印或提供本文档副本的权利仅限于此类软件所适用的许可协议的有效期内。如果该许可因任何原因而终止，您应负责向 CA 书面证明已将本文档的所有副本和部分副本已退还给 CA 或被销毁。

在所适用的法律允许的范围内，CA 按照“现状”提供本文档，不附带任何保证，包括但不限于商品适销性、适用于特定目的或不侵权的默示保证。CA 在任何情况下对您或其他第三方由于使用本文档所造成的直接或间接的损失或损害都不负任何责任，包括但不限于利润损失、投资受损、业务中断、信誉损失或数据丢失，即使 CA 已经被提前明确告知这种损失或损害的可能性。

本文档中涉及的任何软件产品的使用均应遵照有关许可协议的规定且根据本声明中的条款不得以任何方式修改此许可协议。

本文档由 CA 制作。

仅提供“有限权利”。美国政府使用、复制或透露本系统受 FAR Sections 12.212、52.227-14 和 52.227-19(c)(1) - (2) 以及 DFARS Section 252.227-7014(b)(3) 的相关条款或其后续条款的限制。

版权所有 © 2013 CA。保留所有权利。此处涉及的所有商标、商品名称、服务标识和徽标均归其各自公司所有。

CA Technologies 产品引用

本指南参考 CA Spectrum®。

联系技术支持

要获取在线技术帮助以及办公地址、主要服务时间和电话号码的完整列表，请联系技术支持：<http://www.ca.com/worldwide>。

目录

第 1 章： AlarmNotifier 概述	7
关于 AlarmNotifier	7
警报监控过程	8
CA Spectrum Alarm Notification Manager (SANM)	9
第 2 章： 操作 AlarmNotifier	11
启动 AlarmNotifier	11
停止 AlarmNotifier	12
使用进程后台程序启动 AlarmNotifier	13
运行多个 AlarmNotifier	14
AlarmNotifier 输出	15
脚本参数定义	15
永久和陈旧警报	18
第 3 章： 自定义 AlarmNotifier	19
修改脚本	19
在脚本中启用电子邮件通知	19
Windows 平台上的邮件服务	20
限制脚本输出	21
将 CA Spectrum 属性传递给脚本	22
全局警报属性	23
将数据发送到第三方应用程序	25
参数 - USE_NEW_INTERFACE 设为 True	25
参数 - USE_NEW_INTERFACE 设为 False	26
日期和时间	27
自定义 .alarmrc 资源文件	28
.alarmrc 参数	28
重新安装或升级 CA Spectrum	29

第 1 章： AlarmNotifier 概述

此部分包含以下主题：

[关于 AlarmNotifier \(p. 7\)](#)

[警报监控过程 \(p. 8\)](#)

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

关于 AlarmNotifier

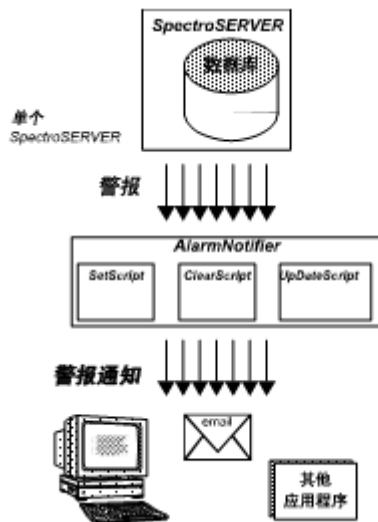
AlarmNotifier 是随核心 CA Spectrum 组件一起安装的 SpectroSERVER 客户端应用程序。AlarmNotifier 应用程序连接到单个 SpectroSERVER，并调用脚本来提供有关 CA Spectrum 警报状态的通知。

从终端 shell 命令提示符下启动 AlarmNotifier。一旦启动后，不管是设置、清除还是更新了警报，它便会持续显示被调用脚本的输出。AlarmNotifier 为 CA Spectrum 提供以下功能：

- 单个 SpectroSERVER 警报监控。
- 生成警报信息的三个脚本：SetScript、ClearScript 和 UpdateScript。
这些脚本包含可以针对您的环境自定义的设置。
- 您可以通过配置资源文件参数来修改 AlarmNotifier 的运行功能。

警报监控过程

AlarmNotifier 增强了 CA Spectrum 警报监控和通知功能。下图说明了 AlarmNotifier 和 CA Spectrum 之间的关联关系：



CA Spectrum 执行一些警报功能，而 AlarmNotifier 执行其他功能。CA Spectrum 轮询已建模的网络元素，并更新 SpectroSERVER 数据库中存储的每个元素的状态信息。

在从网络接收陷阱或检测到网络元素模型中存在关键的状态更改时，CA Spectrum 会生成警报。在 OneClick 拓扑视图中，模型图标由绿色更改为其他颜色指示了警报重要级别。CA Spectrum 在“警报”选项卡中发布有关警报的信息。有关该警报的事件信息显示在 OneClick “内容”面板的“事件”选项卡中。

在 AlarmNotifier 启动时，它会在 CA Spectrum 中注册。然后，会创建一个名为 AlarmNotifier、ClientApp 类型的模型。该模型在任何 CA Spectrum 拓扑视图中都不可见。但是，您可以在“事件”选项卡中看到它。“事件”选项卡显示了与该模型相关的信息，如应用程序的启动和停止时间。

AlarmNotifier 会查询 SpectroSERVER 并请求有关现有警报的信息。AlarmNotifier 可运行脚本并生成有关现有警报的通知。

每次设置、清除或更新警报时，AlarmNotifier 都会从 SpectroSERVER 接收信息并调用相关脚本。AlarmNotifier 脚本可以启动电子邮件通知，以便将警报发送给网络中的人员。还可以将警报信息传送给第三方应用程序。

CA Spectrum Alarm Notification Manager (SANM)

CA Spectrum Alarm Notification Manager (SANM) 是 CA Spectrum 中的一个附加组件，用来增强 AlarmNotifier 的功能。SANM 策略管理员允许您创建多个警报筛选策略，您可以将这些策略与唯一命名的 AlarmNotifier 应用程序实例进行关联。使用这些策略可指示 AlarmNotifier 仅为您认为相关的警报生成通知。

SANM 策略管理员允许您根据需要策略与 AlarmNotifier 应用程序进行关联。您还可以使用排定程序实用工具自动执行此关联过程。

安装 SANM 后，AlarmNotifier 可提供以下功能：

- 分布式 SpectroSERVER 警报监控
- 能提供更多警报信息的新增脚本参数
- 用于确认和清除警报的命令
- 可用于记录 AlarmNotifier 活动以及同时运行多个 AlarmNotifier 实例的其他启动选项
- SANM 警报筛选工具

注意：有关 SANM 的详细信息，请参阅《*Alarm Notification Manager 用户指南*》。

第 2 章：操作 AlarmNotifier

此部分包含以下主题：

[启动 AlarmNotifier](#) (p. 11)

[停止 AlarmNotifier](#) (p. 12)

[使用进程后台程序启动 AlarmNotifier](#) (p. 13)

[AlarmNotifier 输出](#) (p. 15)

[脚本参数定义](#) (p. 15)

[永久和陈旧警报](#) (p. 18)

启动 AlarmNotifier

AlarmNotifier 安装在以下目录中：

<\$SPECROOT>/Notifier

默认情况下，该目录包含以下文件：

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

如果已安装 SANM，则此目录中还可能存在一些其他文件。有关详细信息，请参阅《*Alarm Notification Manager 用户指南*》。

注意：在 Windows 环境中，请在首次启动 AlarmNotifier 之前配置邮件服务。

遵循这些步骤:

1. 确认 `<$SPECROOT>/Notifier/.alarmrc` 资源文件中的 LANDSCAPE 参数指定了要连接的格局。
2. 确认此格局已启动并正在运行。
3. 使用默认资源文件 (.alarmrc) 或修改版本的资源文件启动 AlarmNotifier, 如下所示:
 - 使用默认资源文件启动 AlarmNotifier:
`<$SPECROOT>/Notifier/AlarmNotifier`
 - 使用其他资源文件启动 AlarmNotifier:
`<$SPECROOT>/Notifier/AlarmNotifier -r <资源文件>`

注意: 使用 SANM 时, 还可以随 AlarmNotifier 命令指定一些其他选项。有关详细信息, 请参阅《*Alarm Notification Manager 用户指南*》。

详细信息:

[Windows 平台上的邮件服务 \(p. 20\)](#)

停止 AlarmNotifier

一旦启动, AlarmNotifier 便会持续运行。要停止 AlarmNotifier, 请使用以下命令之一:

停止前台运行的 AlarmNotifier

遵循这些步骤:

Windows/Solaris

输入:

Control-C

停止后台运行的 AlarmNotifier

遵循这些步骤:

Windows

要结束 AlarmNotifier 进程, 请使用 `$SPECROOT/lib/SDPM/kill -TERM <pid>`。

Solaris

输入:

```
kill <选项> <AlarmNotifier PID>
```

使用进程后台程序启动 AlarmNotifier

可以使用 CA Spectrum 进程管理后台程序 `processd` 自动启动 AlarmNotifier 进程。

在 CA Spectrum 安装期间，以及每当系统重新启动时，`processd` 就会自动启动。`processd` 启动后，将自动启动并管理其他进程。

要使 `processd` 能够启动和跟踪 AlarmNotifier 应用程序，请创建一个安装票单文件。

注意：默认情况下，会附带 `AlarmNotifier.idb`。因此，您无需从头开始创建该文件。可以将 `AUTOBOOTSTART` 更改为 `Y`。

遵循这些步骤：

1. 使用文本编辑器，创建一个包含以下内容的文件：

```
# 用于 Alarm Notifier 的 Processd 安装票单
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<CSEX>; //
```

2. 将此文件命名为“AlarmNotifier.idb”并保存到以下目录中：

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

`LOGNAMEPATH` 参数指定 AlarmNotifier 应用程序使用的日志文件的名称和路径。

每次 AlarmNotifier 启动时，都会生成一个新的日志文件，并为前一个日志文件创建备份。但是，SpectroSERVER 只存储最近的两个 AlarmNotifier 日志文件。要保留 AlarmNotifier 的所有日志文件输出，请创建一个脚本，以便在单独的文件中保存日志文件输出。

TICKETUSER 参数的值 <USERNAME> 必须是有效的 CA Spectrum 用户模型的用户名。

必须先运行 SpectroSERVER，然后才能启动 AlarmNotifier。因此，可将 STARTPRIORITY 参数设置为 30，以指示 AlarmNotifier 依赖于 SpectroSERVER。有关详细信息，请参阅《分布式 SpectroSERVER 管理员指南》。

运行多个 AlarmNotifier

CA Spectrum processd 启动时可以启动使用不同通知策略的多个 AlarmNotifier。可以通过为这些 AlarmNotifier 创建所需安装故障单文件来启用此设置。执行以下任务以运行多个 AlarmNotifier：

- 为多个 AlarmNotifier 创建所需安装故障单文件，或复制默认的 ALARMNOTIFIER.idb 文件并相应地重命名这些文件。有关详细信息，请参阅[使用进程后台程序启动 AlarmNotifier](#) (p. 13)。
- 创建安装故障单文件之后，可以更改安装故障单文件的值以启用使用首选通知策略的 AlarmNotifier。可以提供以下值：

APPNAME

指定创建用于实施通知策略的应用程序的名称。有关详细信息，请参阅《Alarm Notification Manager 用户指南》。

AUTOBOOTSTART

在 processd 启动时通知进程后台程序启动进程。将值设置为 Y。

WORKPATH

指定可以在其中找到应用程序的工作路径。

LOGNAMEPATH

为 AlarmNotifier 应用程序指定日志文件的名称和路径。

注意：每个实例的 LOGNAMEPATH 必须是唯一的。

ARGV

指定以下值：

```
$SPECROOT/Notifier/AlarmNotifier<CSEXE> -r alarmrc -n <您为 APPNAME 指定的应用程序名称>
```

注意：如果将 AlarmNotifier 可执行文件复制到其他目录以作为单独实例运行，请相应地自定义 WORKPATH、LOGNAMEPATH 和 ARGV。

AlarmNotifier 输出

只要 AlarmNotifier 检测到 CA Spectrum 中设置、清除或更新了警报，AlarmNotifier 便会调用 SetScript、ClearScript 或 UpdateScript。每个脚本都会生成一个包含有关警报状态信息（设置、清除或更新）的通知并进行显示。每个通知都包含[脚本参数定义](#) (p. 15)中定义的参数。

SetScript

在以下情况下针对警报调用：

- AlarmNotifier 已启动并检测到一个现有警报。除非 .alarmrc 资源文件中的 GET_EXISTING_ALARMS 值设置为 “false”，否则 AlarmNotifier 将调用 SetScript。

默认： true。

- 在 AlarmNotifier 运行时 CA Spectrum 生成了警报。

ClearScript

在清除警报时调用。

UpdateScript

在更新警报时调用。在以下情况下，警报被定义为更新：

- 已为警报分配故障排除者，或更改了故障排除者的名称。脚本中的 RepairPerson 参数就表示该故障排除者的名称。
- 警报的状态发生了更改。在“警报”选项卡中输入有关警报的状态信息。脚本中的 AlarmStatus 参数表示警报的状态。
- 在 CA Spectrum 中确认或取消确认了某警报。
- 处于警报状态的设备中发生了新事件或现有事件发生了更改。

脚本参数定义

您可以更新 AlarmNotifier 脚本的参数来自定义其功能。以下列表描述了可用于 SetScript、ClearScript 和 UpdateScript 的参数。

注意： 在将 SANM 与 AlarmNotifier 结合使用时，还有一些其他的脚本参数可用。有关详细信息，请参阅《*Alarm Notification Manager 用户指南*》。

日期

指定 AlarmNotifier 检测到警报被设置、更新或清除时的日期。

格式: yyyy/mm/dd

时间

指定 AlarmNotifier 检测到警报被设置、更新或清除时的时间。

格式: hh:mm:ss

模型类型

指定其中警报被设置、更新或清除的模型的类型。

模型名称

指定发生了警报设置、更新或清除的模型的名称。如果“模型名称”包含特殊字符，请将其作为环境变量传递给脚本以避免错误。特殊字符是命令 shell 解释为具有具体含义的字符（如“\$”或“*”）。

要将模型名称作为环境变量进行传递，请将“模型名称”的属性 ID (0x1006e) 添加到 .alarmrc 文件中的 EXTRA_ATTRS_AS_ENVVARS 参数值中。有关详细信息，请参阅[将 CA Spectrum 属性传递给脚本](#) (p. 22) 和 [.alarmrc 参数](#) (p. 28)。

警报 ID

指定 CA Spectrum 分配给警报的数字标识符。

全局警报 ID

指定 CA Spectrum 分配给警报的唯一数字标识符。与“警报 ID”不同，“全局警报 ID”不仅在 CA Spectrum 环境内唯一，还可以作为唯一标识符传递给其他环境。使用此值可将唯一标识符传送给第三方软件。默认情况下，“全局警报 ID”会在每个 AlarmNotifier 脚本中被注释掉。删除注释标记 (#) 即可传递此参数。

重要程度

为警报 CA Spectrum 指定重要级别代码：关键（红色）、主要（橙色）、次要（黄色）、维护（褐色）、已抑制（灰色）或初始（蓝色）。

可能原因 ID

指定与警报的可能原因相关联的十六进制标识符。

修复人员

指定在“警报”选项卡中分配给警报的故障排除者。首次分配故障排除者以及其后更新故障排除者时，**AlarmNotifier** 都会调用 **UpdateScript**。

下列情况将决定故障排除者（修复人员）的名称是否会显示在 **SetScript** 和 **ClearScript** 生成的通知中：

- 如果在 **AlarmNotifier** 检测到已设置警报后分配故障排除者，则 **SetScript** 不会显示名称。**UpdateScript** 和 **ClearScript** 会显示名称。
- 如果在 **AlarmNotifier** 检测到该设置（对于 **AlarmNotifier** 启动之前存在的警报）之前分配故障排除者，三个脚本都会显示故障排除者的名称。

注意：有关详细信息，请参阅《操作员指南》。

警报状态

指示 **OneClick** 中警报的状态信息。首次输入状态信息以及其后每次更新该更新时，**AlarmNotifier** 都会调用 **UpdateScript**。状态信息通常会显示在 **SetScript** 和 **ClearScript** 生成的通知中。下列情况除外：

- 如果在 **AlarmNotifier** 检测到已设置警报之后输入状态信息，则 **SetScript** 不会显示该信息。**UpdateScript** 和 **ClearScript** 会显示该信息。
- 如果在 **AlarmNotifier** 检测到该设置（对于 **AlarmNotifier** 启动之前存在的警报）之前输入状态信息，三个脚本都会显示状态信息。

注意：有关详细信息，请参阅《操作员指南》。

SpectroSERVER

指定已经设置、更新或清除警报的 **SpectroSERVER** 的主机名称。

格局

指定从中设置、更新或清除警报的格局的句柄。

模型句柄

指定已为其生成警报的模型的句柄。

模型类型句柄

指定已为其设置、更新或清除警报的模型类型的句柄。

IP 地址

指定已为其设置、更新或清除警报的网络元素的 IP 地址。

安全字符串

指定已为其设置、更新或清除警报的模型的安全字符串。

警报状态

指定警报状态是“现有”还是“新增”。

如果警报是在 AlarmNotifier 启动之前设置的，则警报的状态为“现有”。如果 .alarmrc 资源文件中的 GET_EXISTING_ALARMS 参数设置为 true，AlarmNotifier 会为现有警报调用 SetScript。

如果警报是在 AlarmNotifier 启动之后生成的，则警报的状态为“新增”。如果 AlarmNotifier 的 SpectroSERVER 重新连接到之前连接并出现过警报的格式，警报状态同样为“新建”。

已确认

指定警报是否已确认。

用户可清除

指定用户是否可以清除警报。

设备类型

指定已为其设置、更新或清除警报的模型上的 DeviceType 属性的值。有关详细信息，请参阅《认证用户指南》。

原始警报时间

以 \$DATE 和 \$TIME 格式呈现的警报日期和时间。要获取未格式化的警报时间，您可以引用 \$RAW_ALARM_TIME。

注意：未格式化的警报时间是从 UTC 时间 1970 年 1 月 1 日午夜以来经过的秒数。

永久和陈旧警报

在 SpectroSERVER 停止并重新启动时，已有的警报将继续存在。这些警报是“永久”警报。使用永久警报功能，在 SpectroSERVER 关闭时，CA Spectrum 可以保留与警报相关的信息（如故障排除者分配和状态）。

在某些情况下，警报的根本原因会在 SpectroSERVER 关闭和下次重新启动期间得到解决。该警报仍会出现在 OneClick 的“警报”列表中，但被视为陈旧警报。您可以手动清除所有陈旧警报（也称为“残留警报”）。

但是，陈旧警报信息不会被 SpectroSERVER 转发到 AlarmNotifier。相反，它会生成一个新警报以指示格局中存在陈旧或残留警报，并将该警报发送给 AlarmNotifier。手动清除陈旧警报时，该警报也会从 AlarmNotifier 中清除。在清除最后的陈旧警报时，将向陈旧警报通知发出“清除”命令。

第 3 章： 自定义 AlarmNotifier

此部分包含以下主题：

[修改脚本](#) (p. 19)

[限制脚本输出](#) (p. 21)

[将 CA Spectrum 属性传递给脚本](#) (p. 22)

[将数据发送到第三方应用程序](#) (p. 25)

[自定义 .alarmrc 资源文件](#) (p. 28)

[重新安装或升级 CA Spectrum](#) (p. 29)

修改脚本

您可以修改 AlarmNotifier 脚本来自定义 AlarmNotifier 操作和输出。您可以配置脚本以向指定收件人发送电子邮件通知。还可以自定义脚本，以限制警报转换通知所提供信息的范围，或者与第三方应用程序集成。

在脚本中启用电子邮件通知

每个 AlarmNotifier 脚本都包含两个参数（SENDMAIL 和 VARFORMAIL），您可以配置它们以便为警报的故障排除者启用电子邮件通知。您可以为一个或多个脚本启用电子邮件通知。

注意： 为了保留默认脚本配置设置以避免意外丢失，请对要编辑的默认脚本进行备份。

遵循这些步骤：

1. 导航到 <SPECROOT>/Notifier 目录或保存脚本的目录。
2. 使用文本编辑器打开脚本。
注意： 所有脚本将逐个执行。因此，您可以编辑脚本而无需停止 AlarmNotifier。
3. 将脚本中的 SENDMAIL 参数设为 True。
4. 将 VARFORMAIL 参数设为 RepairPerson。
5. 保存并关闭脚本。

电子邮件将发送给“警报”选项卡中分配给警报的故障排除者。此人必须是产生警报的格局中的授权用户（或其中的用户模型）。

注意：发生警报后，将建立 RepairPerson 的值。因此，无法发送邮件以响应设置操作（使用 SetScript）。但是，如果您还使用了 SANM，则可以配置根据设置操作的结果发送邮件。有关详细信息，请参阅《Alarm Notification Manager 用户指南》。

Windows 平台上的邮件服务

在 Windows 中，使用邮件命令来启用邮件服务，以便 AlarmNotifier 脚本可以发送电子邮件通知。

请在 AlarmNotifier 启动之前执行此步骤。否则，将显示一条错误消息提示您配置邮件服务，然后启动 AlarmNotifier。

邮件命令参数

从终端窗口运行邮件命令时，有几个参数是必需的。这些参数如下所示：

-m

指示返回主机名。返回主机是接收传入邮件的计算机。

-h

指示简单邮件传输协议 (SMTP) 主机，即传出电子邮件发送给其处理的计算机。

-u

指示用户名。

注意：AlarmNotifier 可以将通知发送到寻呼机。建议先将 AlarmNotifier 配置为将通知发送到有效的本地邮件帐户，以测试此配置。然后，您可以重新配置邮件服务以将通知发送到寻呼机。

配置邮件服务

您可以在 Windows 上设置邮件服务，以便 AlarmNotifier 可以发送电子邮件通知。

遵循这些步骤：

1. 咨询您的邮件服务器管理员，以确认正确的邮件命令参数值。
2. 打开 bash shell。

3. 输入以下命令：

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

命令成功完成后，将显示其用法列表。

4. 要验证配置是否完整，请查看 `HKEY_LOCAL_USER/Software/SMail` 下的注册表条目。

现在，`hostname`、`smtphost` 和 `username` 项包含了您在命令字符串中加入的信息。

详细信息：

[邮件命令参数](#) (p. 20)

限制脚本输出

您可以注释掉 `AlarmNotifier` 脚本参数以减少脚本的输出量。

注意： 修改脚本之前，请查看[脚本参数定义](#) (p. 15)中的参数说明。建议您对抑制的信息进行全面深入的了解。

遵循这些步骤：

1. 关闭要配置的 `AlarmNotifier` 应用程序的实例。
2. 导航到默认脚本目录 `<$SPECROOT>/Notifier` 或您要编辑的脚本所在的目录。
3. 使用文本编辑器打开脚本。
4. 在要抑制的每个回显命令行的开头键入一个 `#` 字符，将其注释掉。以下示例中注释掉了 `UserClearable` 参数：

```
echo "安全字符串：    " $SECSTR
echo "警报状态：      " $ALARMSTATE
echo "已确认：" $ACKD
#echo "用户可清除：    " $CLEARABLE
```

该脚本生成通知时，已注释掉的参数相关的信息不会显示在屏幕上。

5. 保存并关闭脚本。

注意： 不要注释掉或修改变量本身或转换命令的赋值。如果更改了这些行，脚本将无法正确显示警报信息。

将 CA Spectrum 属性传递给脚本

可以将具有警报的模型的属性传递给 AlarmNotifier。模型属性可用作 SetScript、ClearScript 或 UpdateScript 中的参数。可以使用 .alarmrc 参数 EXTRA_ATTRS_AS_ENVVARS 或 EXTRA_ATTRS_AS_ARGS 来传递属性。要启用这些参数，请将 USE_NEW_INTERFACE .alarmrc 参数设为 TRUE。

EXTRA_ATTRS_AS_ENVVARS 将属性作为环境变量传递给 AlarmNotifier。EXTRA_ATTRS_AS_ARGS 将属性作为命令行参数进行传递。对于大多数属性而言，可以使用其中任何一种机制。但是，如果换行符或特殊字符可能会使解析额外数据的脚本出现问题，则必须使用 EXTRA_ATTRS_AS_ENVVARS。在 USE_NEW_INTERFACE=TRUE 时，应使用环境变量来传递 \$STATUS、\$EVENTMSG 和 \$PCAUSE 以避免此问题。

在 .alarmrc 文件中，设置与要传递的 CA Spectrum 属性 ID 等效的相应参数。您可以使用十六进制或十进制符号引用属性 ID。

如果使用 EXTRA_ATTRS_AS_ENVVARS 将属性作为环境变量进行传递，请使用下列语法在脚本中引用此变量：

```
$SANM_<attribute_ID>
```

<attribute_ID>

指定要引用的属性的属性 ID。如果已在 .alarmrc 文件中使用十六进制符号调用此属性，则在脚本中也必须使用十六进制符号。如果您已在 .alarmrc 文件中使用十进制符号调用此属性，则也请在脚本中使用十进制符号。

Windows 会自动将环境变量设为大写。因此，引用这些变量时，请使用大写格式，如 \$SANM_0X100C5。

如果使用 EXTRA_ATTRS_AS_ARGS 以参数形式传递属性，您可以在脚本中通过为变量赋值来引用此变量：

```
<variable>=$<x>
```

<variable>

指定用于保存属性值的变量。

<x>

指定适当的变量顺序号和已传递的参数的数目。

示例

下面显示了 .alarmrc 文件中传递到 AlarmNotifier 的四个 CA Spectrum 属性示例。随后可以在脚本中引用这些属性。

.alarmrc 文件引用

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

脚本引用

```
#这些行会将 0x110df 和 0x117dc 分别读入变量 MAC_ADDRESS
#和 FIRMWARE_VERSION。
```

```
shift 9
MAC_ADDRESS=$1
FIRMWARE_VERSION=$2
```

```
#这些行会输出每个属性的值。（Solaris 平台）
echo "属性 0x100c5 的值是：" $SANM_0x100c5
echo "属性 0x11f84 的值是：" $SANM_0x11f84
echo "属性 0x110df 的值是：" $MAC_ADDRESS
echo "属性 0x117dc 的值是：" $FIRMWARE_VERSION
```

```
#这些行会输出每个属性的值。（Windows 平台）
#对环境变量的引用均为大写
echo "属性 0x100c5 的值是：" $SANM_0X100C5
echo "属性 0x11f84 的值是：" $SANM_0X11F84
echo "属性 0x110df 的值是：" $MAC_ADDRESS
echo "属性 0x117dc 的值是：" $FIRMWARE_VERSION
```

详细信息：

[.alarmrc 参数](#) (p. 28)

全局警报属性

本节列出了 CA Spectrum 全局警报的属性及其相应的属性 ID。请按照[将 CA Spectrum 属性传递给脚本](#) (p. 22)中说明的方法，将任意属性传递给 AlarmNotifier。

Acknowledged

0x11f4d

Alarm_Source

0x11fc4

Alarm_Status

0x11f4f

Cause_Code

0x11f50

Cleared_By_User_Name

0x11f51

Creation_Date

0x11f4e

ImpactScope

0x1290e

ImpactSeverity

0x1290d

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

注意：Service_Impact 并不能始终反映服务的当前运行状况。Service_Impact 表示在生成设备警报时服务的运行状况。如果只是服务运行状况发生变化，Service_Impact 并不会更改。

将数据发送到第三方应用程序

您可以自定义或替换 SetScript、ClearScript 或 UpdateScript，以创建与第三方应用程序的集成。

您可以补充和扩展 SetScript、ClearScript 或 UpdateScript 的功能。可以在这些脚本中包括 CA Spectrum CLI 命令，以便从 SpectroSERVER 检索更多信息。还可以在脚本中添加自己的代码，以便将数据发送到第三方应用程序。

如果不想使用现有脚本的功能，可以指示 AlarmNotifier 运行您自己的脚本或可执行文件。在 AlarmNotifier 资源文件 (.alarmrc) 中，Set_Script 参数控制在设置警报时要运行的脚本。Clear_Script 参数控制在清除警报时要运行的脚本，Update_Script 参数控制在更新警报时要运行的脚本。

默认情况下，Set_Script 参数具有 SetScript 的值，Clear_Script 参数具有 ClearScript 的值，Update_Script 参数具有 UpdateScript 的值。您可以修改这些参数的值，以便在设置、清除或更新警报时启动不同的脚本。

自定义脚本参数取决于要从 CA Spectrum 中提取的数据以及要将数据发送到的第三方应用程序。要创建自己的脚本或可执行文件，首先要了解从 CA Spectrum 传递给接收脚本或可执行文件的参数。您的脚本或可执行文件必须按正确的顺序从 CA Spectrum 接收所有参数。

参数 - USE_NEW_INTERFACE 设为 True

下表显示了 USE_NEW_INTERFACE .alarmrc 参数设置为 TRUE 时，传递给每个脚本的每个参数的编号、名称和格式。

在 USE_NEW_INTERFACE 设置为 TRUE 时，Status 参数作为环境变量发送。参数的顺序相应地会受到影响。

参数	名称	格式
1	日期	yyyy/mm/dd
2	时间	hh:mm:ss
3	模型类型	文本
4	模型名称	文本
5	警报 ID	整数
6	重要级别	文本
7	原因	文本
8	修复人员	文本
9	服务器	文本
10	格局	十六进制
11	模型句柄	十六进制
12	模型类型句柄	十六进制
13	IP 地址	xxx.xxx.xxx.xxx
14	安全字符串	文本
15	警报状态	文本
16	已确认	文本
17	可清除	文本
18	设备类型	文本

参数 - USE_NEW_INTERFACE 设为 False

下表显示了 USE_NEW_INTERFACE .alarmrc 参数设置为 FALSE 时，传递给每个脚本的每个参数的编号、名称和格式。

注意：如果随 AlarmNotifier 启用了 SANM，还会传递一些其他参数。有关详细信息，请参阅《Alarm Notification Manager 用户指南》。

参数	名称	格式
1	日期	yyyy/mm/dd

参数	名称	格式
2	时间	hh:mm:ss
3	模型类型	文本
4	模型名称	文本
5	警报 ID	整数
6	重要级别	文本
7	原因	文本
8	修复人员	文本
9	状态	文本
10	服务器	文本
11	格局	十六进制
12	模型句柄	十六进制
13	模型类型句柄	十六进制
14	IP 地址	xxx.xxx.xxx.xxx
15	安全字符串	文本
16	警报状态	文本
17	已确认	文本
18	可清除	文本
19	设备类型	文本

日期和时间

下列条件适用于 Set、Clear 和 Update 脚本中的 Date 和 Time 参数：

- **对于 SET 脚本：** Date 和 Time 从 CREATION_DATE (0x11f4e) 属性派生。在创建警报时，SpectroSERVER 会设置该属性。
- **对于 CLEAR 脚本：** Date 和 Time 从 CLEAR_DATE (0x129af) 属性派生。在清除警报时，SpectroSERVER 会设置该属性。
- **对于 UPDATE 脚本：** Date 和 Time 反映 AlarmNotifier 接收警报已更新通知的时间。SpectroSERVER 不设置此值。不要依赖此值来确定发生更新的准确时间。

自定义 .alarmrc 资源文件

保存在 Notifier 目录中的 .alarmrc 资源文件包括 AlarmNotifier 运行参数。您可以对资源文件进行如下修改：

- 指定 AlarmNotifier 是否处理在 AlarmNotifier 启动之前由 CA Spectrum 生成的警报。
- 指定要传递给 AlarmNotifier 的可选 CA Spectrum 属性。
- 使用自定义脚本替换 SetScript、ClearScript 或 UpdateScript。
- 指定 AlarmNotifier 连接到的 SpectroSERVER。
- 禁用那些指定未计划部署的 AlarmNotifier 操作的参数，以减少网络通信。

注意：注释掉的参数并不会被禁用。而是使用其默认值。

遵循这些步骤：

1. 导航到 <SPECROOT>/Notifier 目录，并对 .alarmrc 文件进行备份。
2. 使用您的首选文本编辑器打开文件。
3. 编辑文件，您可以禁用可选参数或输入新的参数值。禁用参数时，可以为其指定值 False 或将值留空。

注意：不要禁用必需参数，也不要删除任何参数。

4. 保存并关闭文件，然后重新启动 AlarmNotifier。
重新启动 AlarmNotifier 后，您的更改就会生效。

.alarmrc 参数

以下列表介绍了随 AlarmNotifier 一起提供的资源文件参数。有关详细信息，请参阅《Alarm Notification Manager 用户指南》。

LANDSCAPE

标识 AlarmNotifier 连接到的初始 SpectroSERVER 主机。仅在此处输入一个名称。如果 LANDSCAPE 未定义，AlarmNotifier 将默认使用 VNM 格局映射中的第一个格局。信息窗口会显示默认的格局句柄。

VNM_MAIL_TIMEOUT

指定取消请求之前邮件服务等待 SpectroSERVER 响应的最短时间。

默认值：60,000 毫秒（1 分钟）

VNM_CONNECT_TIME_LIMIT

指定 AlarmNotifier 和 SpectroSERVER 之间的初始 TCP 连接请求超时之前的最小延迟。

默认值: 60,000 毫秒 (1 分钟)

KEEP_ALIVE_TIMEOUT

指定保持活动请求超时之前的时间量。

默认值: 30,000 毫秒 (30 秒)

KEEP_ALIVE_INTERVAL

指定发送给 SpectroSERVER 的保持活动请求的间隔时间。保持活动请求会查看 SpectroSERVER 是否仍与 AlarmNotifier 连接。如果 AlarmNotifier 未接收到对请求的响应，它将断开与 SpectroSERVER 的连接。如果您的 SpectroSERVER 对这些请求的响应很慢，您可以加大该值，以防止 AlarmNotifier 与 SpectroSERVER 断开连接。

默认值: 60,000 毫秒 (1 分钟)

SEND_ALARM_DELAY

指定连续警报通知之间的最小延迟。

默认值: 1,000 毫秒 (1 秒)

GET_GRAY_INITIAL_ALARMS

指定您是否希望接收 Gray (灰色) 和 Initial (初始) 警报。如果您不想接收灰色或初始警报，请将此参数设为 FALSE。此设置可减少 AlarmNotifier 产生的网络通信量，改善网络性能。

GET_EXISTING_ALARMS

指定您是否希望接收与调用 AlarmNotifier 时已存在的警报相关的报告。否则，您只会接收 AlarmNotifier 调用后所发生警报的报告。在警报通知的间隔期间，会显示以下消息：“正在等待来自 SpectroSERVER 的更多警报”。

重新安装或升级 CA Spectrum

重新安装 CA Spectrum 或升级 CA Spectrum 的版本时，安装过程会自动将 SetScript、UpdateScript 和 ClearScript 保存到备份目录中。您已采用其他名称保存的默认脚本的版本（如 SetScript_version1 或 UpdateScript_modified）将保留在 <\$SPECROOT>/Notifier 目录中。该目录还包含重新安装或升级时包含的默认脚本。

此外，安装过程还会将您的 .alarmrc 文件保存到备份目录中。您已采用其他名称保存的 .alarmrc 资源文件的版本（如 .alarmrc1 或 .alarmrc2）将保留在 <\$SPECROOT>/Notifier 目录中。

备份脚本和备份 .alarmrc 会保存到以下目录中：

<\${SPECROOT}/Install-Tools/SAVES_<日期>/<时间>/Notifier

