第 7 章 运行 Nagios 的基本操作

7.1. 验证配置文件的正确性

每次修改过你的配置文件，你应该运行一次检测程序来验证配置的正确性。在运行你的 Nagios 程序之前这是很重要的，否则的话会导致 Nagios 服务因配置的错误而关闭。

为验证你配置，运行 Nagios 带命令行参数 -v，象这样：

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

如 果你确实忘记了一些重要的数据或是错误地配置了，Nagios 将会给出一个报警或是一个错误信息，其中会给出错误的位置。错误信息通常会打印出错误配置的 文件中的那一行。在错误时，Nagios 通常是在预检查出有问题打印出问题的源配置文件行后退回到命令行状态。这使得 Nagios 不会因一个错误而落入需 要验证一个因错误而嵌套的配置循环错误之中。报警信息可**通常**是被忽略的，因为一般那些只是建议性的并非必须的。

一旦你已经验证了你配置文件并修改过你的错误，就可以继续下去，启动或重启 Nagios 服务了。

7.2. 启动与停止 Nagios

有多于一种方式来启动、停止和重启动 Nagios，这里在有更通常做的方式...

提示

在你启动或重启动你的 Nagios 程序之前，你总是要确保你验证你的配置文件已经通过。

7.2.1. 启动 Nagios

1. 初始化脚本：最简单的启动 Nagios 守护进程的方式是使用初始化脚本，象这样：

   ```
   /etc/rc.d/init.d/nagios start
   ```

2. 手工方式：你可以手动地启动 Nagios 守护进程，用命令参数-d，象这样：

```
/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

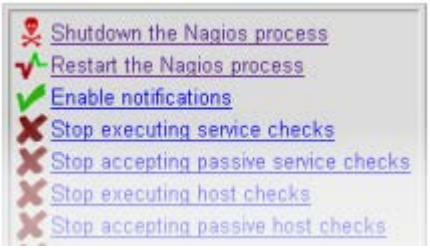
7.2.2. 重启动 Nagios

当你修改了配置文件并想使之生效的话，重启动或重载入动作是必须的。

1. 初始化脚本：最简单地重启动 Nagios 守护进程的方式是使用初始化脚本，象这样：

   ```
   /etc/rc.d/init.d/nagios reload
   ```

2. Web 接口方式：你可以利用 WEB 接口，通过点击“进程信息”的超链接页面里的“重启动 Nagios 进程”来重启动 Nagios，见图



3. 手工方式：你可以手动地发一个 SIGHUP 信号，象这样：

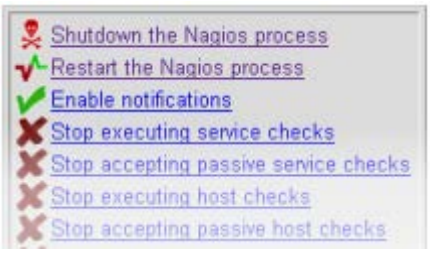
   ```
   kill -HUP <nagios_pid>
   ```

7.2.3. 停止 Nagios

1. 初始化脚本：最简单地停止 Nagios 守护进程的方式是通过初始化脚本，象这样：

   ```
   /etc/rc.d/init.d/nagios stop
   ```

2. Web 接口方式：你可以利用 WEB 接口，通过点击“进程信息”的超链接页面里的“关闭 Nagios 进程”来停止 Nagios，见图



3. 手工方式：你可以手动发一个 SIGTERM 信号，象这样：

   ```
   kill <nagios_pid>
   ```

7.3. 快速启动选项

7.3.1. 介绍

只有很少几件事可以减少 Nagios 的启动或重启总时间。加速启动方法包括有移除些负担还包括加快配置文件处理过程。

利用这些技术在如下一种或几种情况时特别有效：

1. 大型安装配置
2. 复杂地配置(过度地利用模板特性)
3. 需要进行频繁重启动的安装模式

7.3.2. 背景

每次 Nagios 启动和重启时，在它着手进行监控工作之前必须要处理配置文件。启动过程中的配置处理包括如下几步：

1. 读入配置文件
2. 解析模板定义
3. 重粘连("Recombobulating")对象(是我想到的应做各种工作)
4. 复制对象定义
5. 继承对象属性
6. 对象定义排序
7. 验证对象关联关系的完整性
8. 验证回路
9. 和其他...

当有很大的或是很复杂的配置文件要处理时有几步非常消耗时间的。有没有加快这些的办法？当然有！

7.3.3. 评估启动时间

在做让启动速度更快的事情之前，需要看看可能性有多少和是否有必要涉足此事。这个比较容易－只是用-s 命令行开关启动 Nagios 以取得计时和调度信息。

下面是个输出样例(做过精减，只是显示了有关部分)，在这个例子中，假定 Nagios 配置为对 25 个主机和超过 10,000 个服务进行监控。

```
/usr/local/nagios/bin/nagios -s /usr/local/nagios/etc/nagios.cfg

Nagios 3.0-prealpha
```

Last Modified: 01-27-2007

License: GPL

Timing information on object configuration processing is listed below. You can use this information to see if precaching your object configuration would be useful.

Object Config Source: Config files (uncached)

OBJECT CONFIG PROCESSING TIMES (* = Potential for precache savings with -u option)

Read:	0.486780 sec	
Resolve:	0.004106 sec	*
Recomb Contactgroups:	0.000077 sec	*
Recomb Hostgroups:	0.000172 sec	*
Dup Services:	0.028801 sec	*
Recomb Servicegroups:	0.010358 sec	*
Duplicate:	5.666932 sec	*
Inherit:	0.003770 sec	*
Recomb Contacts:	0.030085 sec	*
Sort:	2.648863 sec	*
Register:	2.654628 sec	

```
Free:                   0.021347 sec

                        ============

TOTAL:                  11.555925 sec  *= 8.393170 sec (72.63%) estimated
savings



Timing information on configuration verification is listed below.


CONFIG VERIFICATION TIMES          (* = Potential for speedup with -x
option)

---------------------------------

Object Relationships: 1.400807 sec

Circular Paths:       54.676622 sec  *

Misc:                 0.006924 sec

                      ============

TOTAL:                56.084353 sec  *= 54.676622 sec (97.5%) estimated
savings
```

OK，看看发生了什么。先看汇总信息，大概有 11.6 秒用于处理配置文件有 56 秒来验证配置。这意味着每次用这个配置启动或重启 Nagios 时，它大约会有 68 秒来做启动事项而不会做任何监控的事情！如果是在定制配置 Nagios 过程中也是不可容忍的。

那么怎么办？看一下输出内容，如果运用了优化选项，Nagios 将可以在配置读取过程节省大约 8.4 秒而在验证过程可节省 63 秒。

哇！从 68 秒到只有 5 秒？！是的！看看下面是怎么做到的。

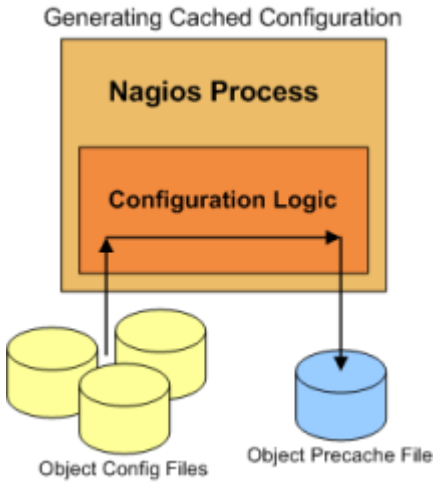
7.3.4. 预缓存对象配置

Nagios 可在解析配置文件过程中做些加速，特别是当配置中使用了模板来做继承等的时候。为降低 Nagios 解析配置文件的处理时间可用 Nagios 预处理与预缓存配置文件的功能。

当用-p 命令参数来运行 Nagios 时，Nagios 将读入配置文件，处理后将配置结果写入预缓存文件(由主配置文件中 precached_object_file 域指定文件位置)。该预缓存配置文件将包含了预处理后的信息将使 Nagios 处理配置文件更容易和快捷。必须把-p 参数选项与-v 或-s 命令参数一起使用，如下例。注意要做预缓存配置文件之前配置应是已被验证过的。

```
/usr/local/nagios/bin/nagios -pv /usr/local/nagios/etc/nagios.cfg
```

预缓存配置文件有大小明显地比原有配置文件大。这是正常的由设计初衷决定的。

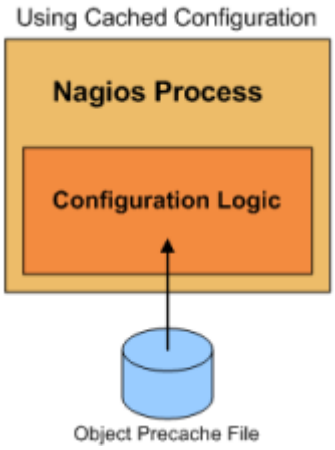


一旦预缓存对象配置文件创建，可以启动 Nagios 时带上-u 命令行选项以让它使用预缓存配置文件而不是配置文件本身。

```
/usr/local/nagios/bin/nagios -ud /usr/local/nagios/etc/nagios.cfg
```

重要

如果更改了配置文件，必须在 Nagios 重启动前要重新验证和重建预缓存配置文件。如果没有重构建预缓存配置文件，Nagios 将使用旧配置运行因为是由旧配置生成的预缓存文件，而不是用新的原始配置文件。



7.3.5. 跳过回路检测

第二步(也是最耗时)部分是对配置中的回路进行检测。在上面例子中这一步几乎用去了 1 分钟来验证配置验证。

什 么时回路检测和为什么要做这么长时间？回路检测逻辑是为了确保在你的主机、主机依赖、服务和服务依赖等对象之间不存在任何的循环路径。如果在配置中有循环 路径，Nagios 将会因死锁而停止。用时较长原因是由于没有使用较高效的算法。欢迎提供更高效发现回路的算法。提示：这意味着 EMail 给我有关 Nagios 论文的计算机科学系研究生将有机会得到些回赠代码。:-)

如果你想在 Nagios 为启动时跳过回路检测，可以在命令行回加上-x 参数，象这样：

```
/usr/local/nagios/bin/nagios -xd /usr/local/nagios/etc/nagios.cfg
```

重要

当要在启动和重启前跳过回路检测之前，验证配置文件的正确性是非常非常重要的！没有这么做将有可能导致 Nagios 逻辑上的死锁。你已被我提醒过了啊！

7.3.6. 联合起来使用

按照下面步骤将会使用预缓存配置文件并且跳过回路检测以充分加速启动。

1、验证配置文件并生成预缓存配置文件，用如下命令：

```
/usr/local/nagios/bin/nagios -vp /usr/local/nagios/etc/nagios.cfg
```

2、如果 Nagios 正在运行，停掉它；

3、启动 Nagios，让其使用预缓存配置文件而且跳过回路检测：

/usr/local/nagios/bin/nagios -uxd /usr/local/nagios/etc/nagios.cfg

4、 当更改了原始配置文件时，需要重启动 Nagios 并修改现有内容，重新回到步骤 1 去验证配置并重构预缓存配置文件。一旦做好了，就可以通过 Web 接口来重 启 Nagios 或是在系统中发个 SIGHUP 信号，如果没有重构预缓存配置文件，Nagios 将用旧配置运行，因为它首先会读入缓存配置文件而不是源配置 文件；

5、就这么多！祝你可以加快启动过程。

7.4. 关于 CGI 程序模块的信息

7.4.1. 说明

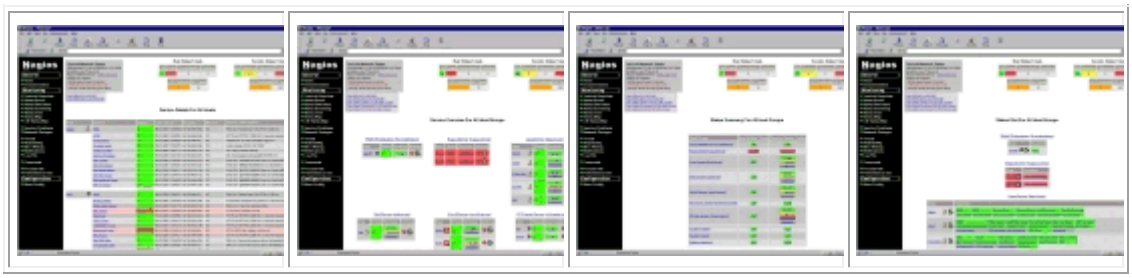
这里将描述一下随 Nagios 发行的几个 CGI 程序模块，每个 CGI 模块都需要做充分的授权设置。默认情况下 CGI 程序将依依赖于你在 Web 服务程序里的授权和对你所请求的视图给你的授权。更多的有关授权配置的信息可以在这里找到。

7.4.2. 索引

- Status CGI
- Status map CGI
- WAP interface CGI
- Status world CGI (VRML)
- Tactical overview CGI
- Network outages CGI
- Configuration CGI
- Command CGI
- Extended information CGI
- Event log CGI
- Alert history CGI
- Notifications CGI
- Trends CGI
- Availability reporting CGI
- Alert histogram CGI
- Alert summary CGI

Status CGI

表 7.1.



模块文件名 *status.cgi*

描述：在 Nagios 里这是一个很重要的 CGI 模块。它可以让你观测到被监测的全部主机和服务的当前状态。它将生成本个主机类型的输出报告 — 全部的（或部分主机）以成组方式给出状态报告和全部的服务（或部分主机上的全部服务）的状态。

授权要求：

1. 如果你已被**授权对全部主机**你就可以看到全部主机和全部服务。
2. 如果你已被**授权对全部的服务**你就可以看到全部服务。
3. 如果你是一个被**授权的联系人**你就可以看到以你为联系人的全部主机和服务。

Status Map CGI

模块文件名 *statusmap.cgi*

描述：这个 CGI 模块将创建一个基于你监测网络全部主机的二维地图。使用 Thomas Boutell 的 gd 库(版本是 1.6.3 或更高)来生成一个 PNG 图，里面的二维坐标依赖于每个主机对象的定义（包括可以给每个主机定义一个好看的图标）。如果你宁可让 CGI 程序自己自动地设定主机的坐标，用一下这个 default_statusmap_layout 域来指定一个二维图生成算法。

授权要求：

1. 如果你已被**授权对全部主机**你就可以看到全部主机。
2. 如果你是一个被**授权的联系人**你就可以看到以你为联系人的主机。

注意

没有被授权的用户只能看到那些主机的节点处于**未知**状态。我真的让它无法看到**任何**东西，如果你无法看到主机依赖的话，你甚至无法看到一个二维图...

WAP Interface CGI

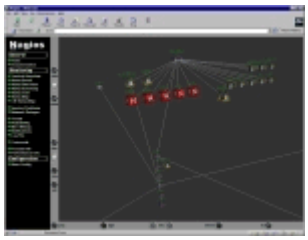
模块文件名 *statuswml.cgi*

描述：这个 CGI 模块将给 WAP 接口提供网 络状态服务。如果你有一个 WAP 设备(象一个带因特网接入能力的移动电话)，你可以在移动中观看状态信息。在主机组汇总、主机概览、主机详细信息、服务详 细信息、全部的故障告警、全部未处理故障等等不同的报告，除了状态信息外，同样可以从移动电话里来设置取消告警、关闭检测和通知故障等。这个功能很酷吧？

授权要求：

1. 如果你已被**授权看系统信息**你可以看到 Nagios 进程信息。
2. 如果你已被**授权对全部主机**你可以看到全部主机和服务的状态数据。
3. 如果你已被**授权对全部的服务**你可以看到全部服务的状态数据。
4. 如果你是一个被**授权的联系人**你可以看到以你做为联系人的主机和服务的状态数据。

Status World CGI (VRML)



模块文件名 *statuswrl.cgi*

描述：这个 CGI 模块将对你所监控网络的全部主机生成一个三维虚拟视图。这些绘制中所用的主机三维坐标(以及渲染图片)来自于配置文件中的主机定义。如果

你想让 CGI 程序模块自动地生成三维坐标，可以设置 default_statuswrl_layout 域来指定一个三维图坐标生成算法。同样，在你要做观察之前你也应在你系统里安装一个虚拟现实的浏览器(象 Cortona、Cosmo Player 或 WorldView)。

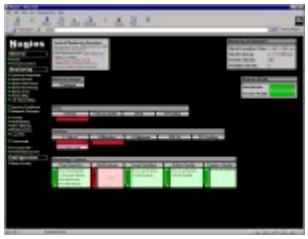
授权要求：

1. 如果你已被**授权对全部主机**你就可以看到全部主机。
2. 如果你是一个被**授权的联系人**你就可以看到以你为联系人的主机。

注意

对于没有被授权的用户，将在没授权的主机节点上看到**未知**状态。我真的让他无法看到**任何**东西，如果你无法看到主机依赖关系时你甚至无法看到一个三维图...

Tactical Overview CGI



模块文件名 *tac.cgi*

描述：这个 CGI 模块给了一个网络活动的 “鸟瞰图”。这容许你快速地得到网络概况、主机状态和服务状态。在已被“处理”的故障(象被认同的和关闭告警的故障)和没有被捕获的问题之间做出区分辨 别，且是需要提请关注的。如果你在监控大量的主机和服务并且想只是用一组画面来分析处理这些故障的话这个会很有用。

授权要求：

1. 如果你已被**授权对全部主机**你可以看到全部主机和全部服务。
2. 如果你已被**授权对全部的服务**你可以看到全部服务。
3. 如果你是一个被**授权的联系人** 你就可以看到以你为联系人的全部主机和服务。

Network Outages CGI

模块文件名 *outages.cgi*

描述：这个 CGI 将给出你网络中的引发网络出错的“问题”主机列表。这对于管理一个大型的网络和想快速定位网络故障来源的情况是很有用的。列表中的主机将按出错问题的先后关系来排列。

授权要求：

1. 如果你已被**授权对全部主机**你就可以看到全部主机。
2. 如果你是一个被**授权的联系人**你就可以看到以你为联系人的主机。

Configuration CGI

模块文件名 *config.cgi*

描述：这个 CGI 模块将让你可以看到全部对象(象主机、主机组、联系人、联系人组、时间周期、服务等等)的配置，这些配置写在你的对象配置文件里面。

授权要求：

1. 你必须被**授权可以看到任何配置**信息和任意一种配置内容。

Command CGI

模块文件名 *cmd.cgi*

描述：这个 CGI 模块将让你给 Nagis 进程发出命令。虽然它有很多个命令参数，但你最好是独立地使用它们。在不同的 Nagios 版本间它们有很大地不同。用 extended information CGI 模块来做为发布命令的起点。

授权要求：

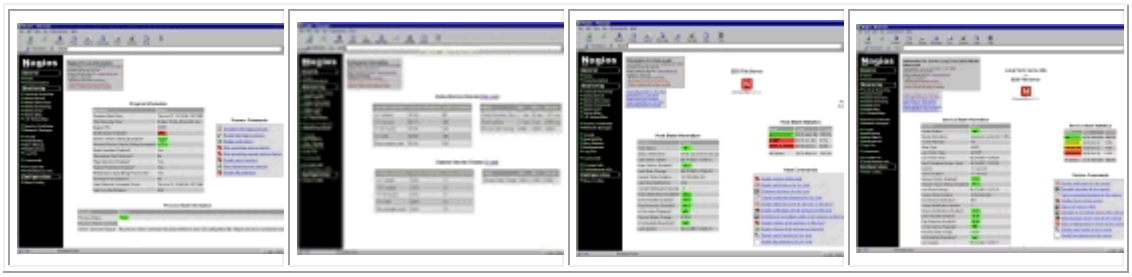
1. 你必须被**授权做系统命令**以使你发出对 Nagios 有影响的命令(重启动、关闭、模式切换等等)。
2. 如果你被**授权在全部主机上执行命令**以使你可以对全部主机和服务发出命令。
3. 如果你被**授权对全部服务执行命令**以使你可对全部服务发出命令。
4. 如果你是一个被**授权的联系人**你可以对你做为联系人的主机和服务上发出命令。

注意

如果*没有*使用在 CGI 配置文件里 use_authentication 选项，这个 CGI 模块将**不会**让你对 Nagios 执行任何命令，这是对你设置的一种保护。如果你决定在 WEB 里设置成非授权管理状态来运行，我建议你最好移走这个 CGI 模块。

Extended Information CGI

表 7.2.



模块文件名 *extinfo.cgi*

描述：这个 CGI 模块将让你看到 Nagios 进程信息、主机和服务状态统计、主机和服务注释和其他信息等。同样它也可以做为对 Nagios 发出命令的服务，跟 command CGI 模块一样。虽然它有几个命令参数，但你最好是独立地用它们 — 在不同的 Nagios 版本之间它们会有不同。你可以通过点击在页面边上的“网络健康状况”和“进程信息”里的链接来进到这个 CGI 模块，也可以通过点击 status CGI 里的主机或服务上的链接进入。

授权要求：

1. 你必须被**授权看系统信息**以使你可以看到进程信息报告。
2. 如果你已被**授权对全部主机**你可以看到全部主机和服务的扩展信息。
3. 如果你已被**授权对全部的服务**你可以看到全部服务的扩展信息。

4. 如果你是一个被**授权的联系人**你可以看到以你做联系人的全部主机与服务的扩展信息。

Event Log CGI

模块文件名 *showlog.cgi*

描述：此 CGI 模块用于显示日志文件。如果已设置日志回滚使能，可以用顶部的导航链接来在打包的日志文件中浏览当前告警。

授权要求：

1. 你必须被**授权看系统信息**以使你可看到日志文件报告。

Alert History CGI

模块文件名 *history.cgi*

描述：这个 CGI 模块被用于显示部分或是全部主机的历史故障。这个是显示日志文件 CGI 模块信息的子集。你可以过滤显示输出内容，只挑出指定类型的故障来查看(如按硬故障和软故障分类，或按服务和主机告警的类型来显示等)。如果你设置了日志回滚，你可以通过页面顶端的导航链接来在打包的日志文件中查看当前的历史信息。

授权要求：

1. 如果你已被**授权对全部主机**你可以看到全部主机和服务的历史信息。
2. 如果你已被**授权对全部的服务**你可以看到全部服务的历史信息。
3. 如果你是一个被**授权的联系人**你可以看到以你做为联系人的全部服务和主机的历史信息。

Notifications CGI

模块文件名 *notifications.cgi*

描述：这个 CGI 模块可以用于显示给各类联系人而发出主机和服务的通知。这个输出是 The output is basically a subset of the information that is displayed by the 日志 CGI 模块显示内容的子集。你可以过滤输出显示内容，只是显示指定的通知类型(如服务通知、主机通知、给指定联系人的通知等)。如果设置了日志回滚选项使能，你可以通过在页面顶端的导航链接来在打包的日志文件中查看当前的通知。

授权要求：

1. 如果你已被**授权对全部主机**你可以查看全部的主机和服务的通知报告。
2. 如果你已被**授权对全部的服务**你可以查看全部服务的通知。
3. 如果你是一个被**授权的联系人**你可以查看以你为联系人的全部服务和主机的通知报告。

Trends CGI

模块文件名 *trends.cgi*

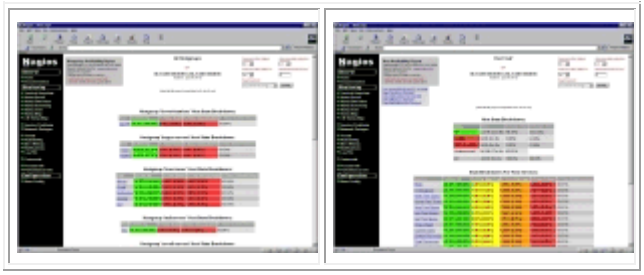
描述：这个 CGI 模块可以创建一个主机或服务的任意时间段内的状态趋势图。为了让此 CGI 模块更有用，你需要设置日志回滚选项使能并保留好打包的日志文件，打包日志文件保留路径在 log_archive_path 域里设置。这个 CGI 模块使用了 Thomas Boutell 的 gd 库(版本 1.6.3 或更高)以创建状态趋势图。

授权要求：

1. 如果你已被**授权对全部主机**你可以查看全部主机和全部服务的趋势图
2. 如果你已被**授权对全部的服务**你可以查看全部服务的趋势图。
3. 如果你是一个被**授权的联系人**你可以查看以你为联系人的全部服务和主机的趋势图。

Availability Reporting CGI

表 7.3.



模块文件名 *avail.cgi*

描述：这个 CGI 模块可用于查看用户定制的指定时间段内的可用性报告。为使这个 CGI 程序更多地被运用，你要设置日志回滚使能并保留打包的日志文件，日志文件保存于 log_archive_path 域里面。

授权要求：

1. 如果你已被**授权对全部主机**你可以查看全部主机和全部服务的可用性数据报告。
2. 如果你已被**授权对全部的服务**你可以查看全部服务的可用性数据报告。
3. 如果你是一个被**授权的联系人**你可以查看以你为联系人的全部服务和主机的可用性数据报告。

Alert Histogram CGI

模块文件名 *histogram.cgi*

描述：这个 CGI 模块可用于显示在用户定制的时间段内的主机和服务的可用性曲线。为使这个 CGI 更多地利用，你须设置日志回滚选项并保留你的打包日志文件，日志文件保存于 log_archive_path 域设置的路径里。这个 CGI 模块使用了 Thomas Boutell 的 gd 库(版本 1.6.3 或更高)以创建历史曲线图。

授权要求：

1. 如果你已被**授权对全部主机**你可以查看全部的主机和全部服务的历史曲线。
2. 如果你已被**授权对全部的服务**你可以查看全部服务的历史曲线。

3. 如果你是一个被**授权的联系人**你可以查看以你为联系人的全部服务和主机的历史曲线报告。

Alert Summary CGI

模块文件名 *summary.cgi*

描述：这个 CGI 模块提供了有关主机和服务告警的概要性的报告，包括总的和最大的告警源等。

授权要求：

1. 如果你已被**授权对全部主机**你可以查看全部主机和全部服务的汇总信息。
2. 如果你已被**授权对全部的服务**你可以查看全部服务的汇总信息。
3. 如果你是一个被**授权的联系人**你可以查看以你为联系人的全部服务和主机的汇总信息。