

## 使用 HostMonitor 实现运维自动化监控

## 目 录

使用 HostMonitor 实现运维自动化监控.....	1
1 前言.....	6
2 程序介绍.....	6
3 操作界面.....	7
4 入门.....	7
4.1 Ping 在线状态检测.....	7
5 常规监控.....	9
5.1 监控 TCP 端口状态 .....	9
5.2 监控 HTTP、HTTPS、FTP 服务器.....	9
5.3 监控 NTP 服务.....	11
5.4 监控 SSL 数字证书有效期 .....	11
6 制定时间计划.....	12
6.1 设置假期.....	15
7 RMA 远程监控代理.....	16
7.1 Windows 平台 .....	16
7.2 Linux 及类 Unix 平台.....	18
7.3 在 HostMonitor 中使用 RMA.....	19
8 个性化告警行为.....	21
8.1 声光报警.....	22
8.2 执行动作.....	23
9 高级监控.....	24
9.1 执行 SQL 语句并判断结果 .....	24
9.1.1 监控 Oracle 数据库会话数 .....	28
9.2 Windows 磁盘可用空间 .....	29
9.3 执行 Shell 脚本并判断结果.....	32
9.3.1 监控 Linux 磁盘可用空间 .....	32
9.4 监控 NFS 挂载状态 .....	36
9.5 监控日志文件内容.....	37



9.6	监控进程数量.....	38
9.6.1	程序退出后自动重启.....	38
10	RCC 远程管理 HostMonitor .....	40
11	后记.....	42
12	附录: HostMonitor 测试方法介绍.....	43
12.1	Ping.....	43
12.2	Trace.....	43
12.3	URL .....	43
12.4	HTTP .....	43
12.5	SOAP/XML .....	43
12.6	Certificate expiration 数字证书有效期.....	44
12.7	Domain expiration 域名有效期.....	44
12.8	SMTP .....	44
12.9	POP3.....	44
12.10	IMAP.....	44
12.11	E-Mail 电子邮件内容 .....	45
12.12	MailRelay.....	45
12.13	TCP .....	45
12.14	UDP .....	45
12.15	NTP.....	45
12.16	DNS.....	45
12.17	DHCP .....	46
12.18	LDAP 目录服务器.....	46
12.19	RADIUS .....	46
12.20	DICOM.....	46
12.21	RAS .....	46
12.22	UNC 共享资源可用性 .....	46
12.23	Drive Free Space 磁盘可用空间 .....	47
12.24	Folder/File Size 文件夹或文件大小.....	47



12.25	Count Files 计算文件数量.....	48
12.26	Folder/File Availability 文件夹或文件可用性 .....	48
12.27	File Integrity 文件完整性 .....	48
12.28	Text Log 文本日志内容 .....	48
12.29	Compare Files 文件比较.....	49
12.30	Process 进程 .....	49
12.31	Service 服务 .....	49
12.32	NT Events Log 事件日志 .....	50
12.33	CPU Usage CPU 利用率 .....	50
12.34	Memory 可用内存.....	50
12.35	Performance Counter 性能计数器.....	51
12.36	WMI.....	51
12.37	Registry 注册表.....	51
12.38	Dominant Process 主要进程 .....	51
12.39	VM host status 虚拟主机状态 .....	51
12.40	VM host CPU usage 虚拟主机 CPU 利用率 .....	52
12.41	VM host free memory 虚拟主机内存 .....	52
12.42	VM host free datastore 虚拟主机可用空间 .....	52
12.43	VM guest status 虚拟机状态 .....	52
12.44	VM guest CPU usage 虚拟机 CPU 利用率 .....	52
12.45	VM guest free memory 虚拟机可用内存 .....	52
12.46	VM guest free disk space 虚拟机可用磁盘空间 .....	52
12.47	Database Server 数据库服务器 .....	53
12.48	ODBC Query 执行 SQL 查询 .....	53
12.49	SNMP Get 获取 SNMP 信息 .....	54
12.50	SNMP Trap SNMP 陷阱 .....	54
12.51	SNMP Table 批量获取 SNMP 信息 .....	55
12.52	Traffic Monitor 网络流量监控 .....	55
12.53	Temperature Monitor 温度监控器 .....	55

12.54	Active Script 活动脚本 .....	55
12.55	Shell Script Shell 脚本 .....	56
12.56	External 外部测试 .....	56
12.57	SSH .....	56
12.58	HM Monitor .....	57

## 1 前言

系统运维工作的挑战性越来越强。一方面运维工作越来越繁重，一是软件日趋复杂化，正常运行所依赖的内外部条件也日益增多，必须要及时有效地检测和判断这些条件是否持续得到满足才能够快速发现和解决问题，才能确保系统平稳运行，二是客观环境对系统运维水平的要求越来越高，客户总是希望系统永远可用，最好永不发生问题，万一出现问题，处理的时间最好也以分钟甚至是秒为单位计算；另一方面运维人员的增长却是有限的。因此系统运维管理人员必须要借助功能强大的自动化监控软件，才能从日常工作中解脱出来，同时提高系统运维水平，第一时间发现和解决问题，而不是等到用户反馈上来才知道系统发生故障。

HostMonitor 就是一款这样的软件，它的监控方法非常丰富，涵盖了日常运维所需的方方面面，尤其是结合其他手段足以实现业务层面的监控，同时提供多用户管理，支持日程计划和自定义动作，是实现自动化运维的一款利器。

本文以 9.90 版本为例进行介绍。

## 2 程序介绍

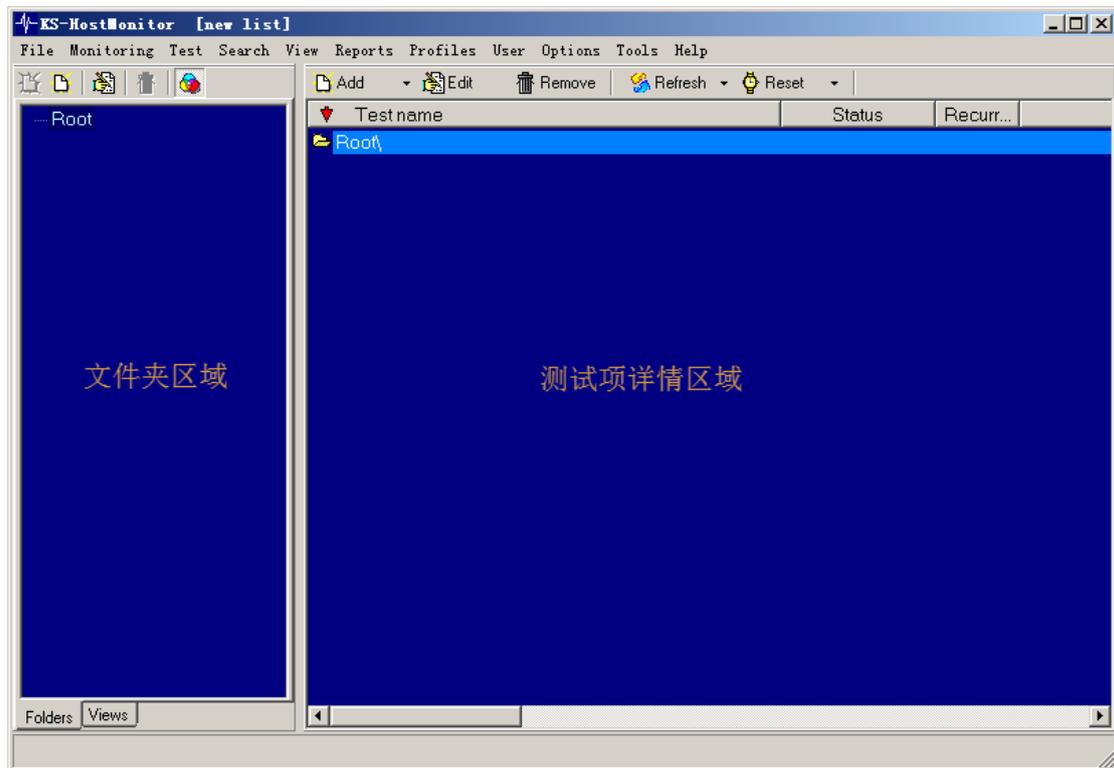
HostMonitor 安装后菜单中的组件较多，通常需要使用的是其主程序即“HostMonitor”。常用的程序还包括：

**RCC:** 远程管理 HostMonitor，可以多用户同时操作。

**RMA Manager:** 远程监控代理（RMA）管理器。

**WatchDog:** 看门口程序，用于监控 HostMonitor 自身状态。

## 3 操作界面



HostMonitor 的主界面很常规，主要包括了传统的上方菜单栏，以及下方的左侧文件夹区域和右侧测试项详情区域。

文件夹区域：可建立多级子文件夹，方便对测试项进行管理。

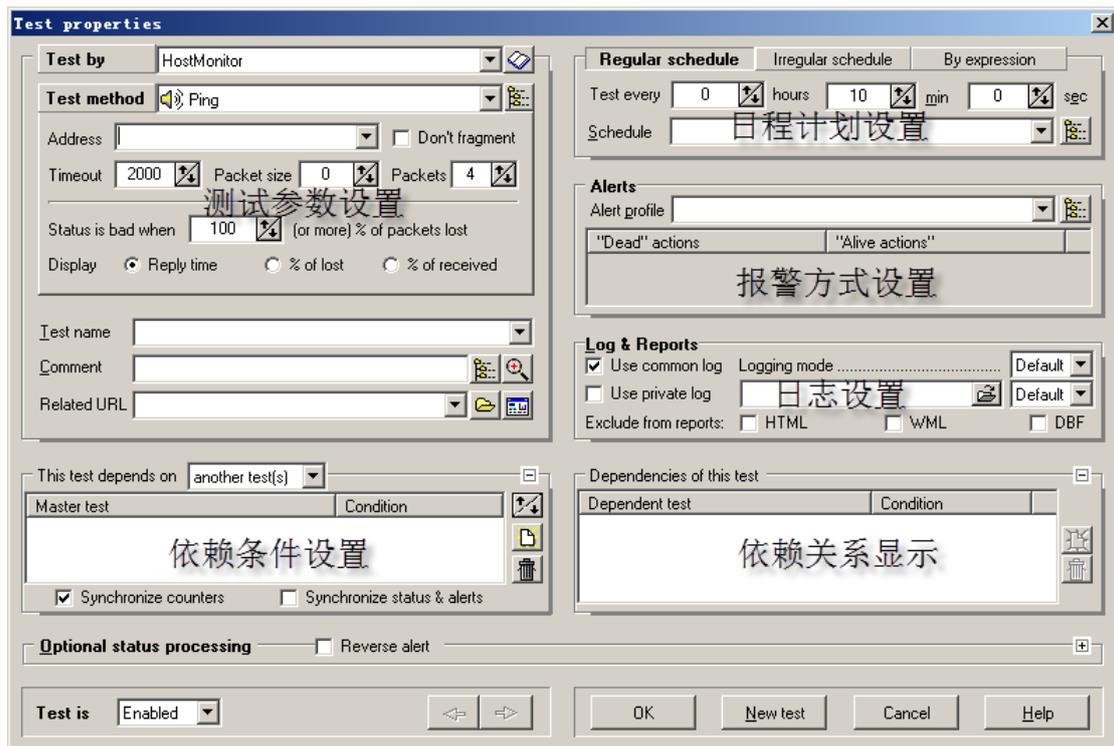
测试项详情区域：显示各项测试的状态等详细信息。提供快捷按钮对测试项进行增改删除。

## 4 入门

### 4.1 Ping 在线状态检测

Ping 是最基本的服务器和网络存活状态检测手段，我们以 Ping 为例，说明 HostMonitor 的基本使用方法。

首先点击 Add，打开测试项添加窗口：



不同区域依次为：

测试参数设置：设置此项测试的基本属性，与测试项所属的类型密切相关。

日程计划设置：选择此项测试的时间间隔，最小为 1 秒。同时也可以设置日程计划，只在需要的时间开启。

报警方式设置：通常选择“Message,Sound”，声光报警，也可以执行其他操作。

日志设置：一般无需修改。

依赖条件设置：设置此项测试需要依赖于哪些测试的结果，一般无需设置。

依赖关系显示：显示有哪些测试项依赖于此项测试。

HostMonitor 提供了数十种内置测试方法（Test method），除了测试参数设置部分存在不同之外，其他设置内容基本都相同。

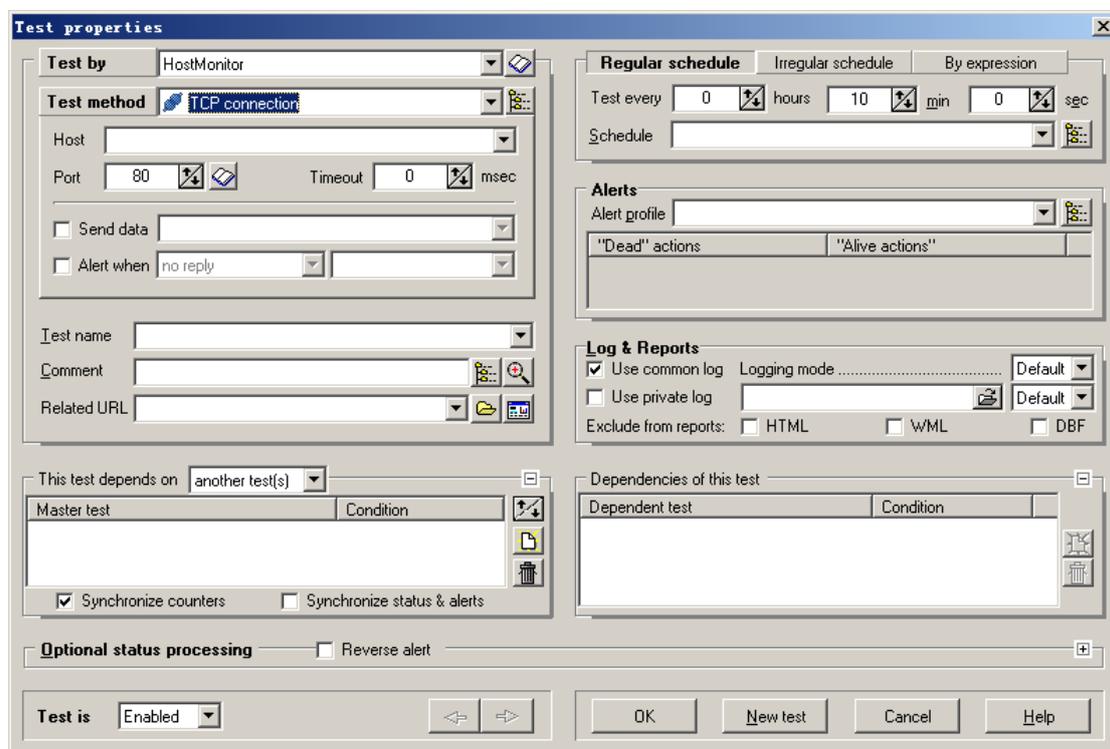
这里我们直接在 Address 栏填入目标 IP 地址“127.0.0.1”，Test name 填入“Ping 本机”，测试间隔改为每分钟 1 次，Alert profile 选择“Message, Sound”即声光报警，然后点击 OK，即完成了第一个监控项的创建。也可以将目标 IP 改为不存在的项目，以测试报警效果。

## 5 常规监控

### 5.1 监控 TCP 端口状态

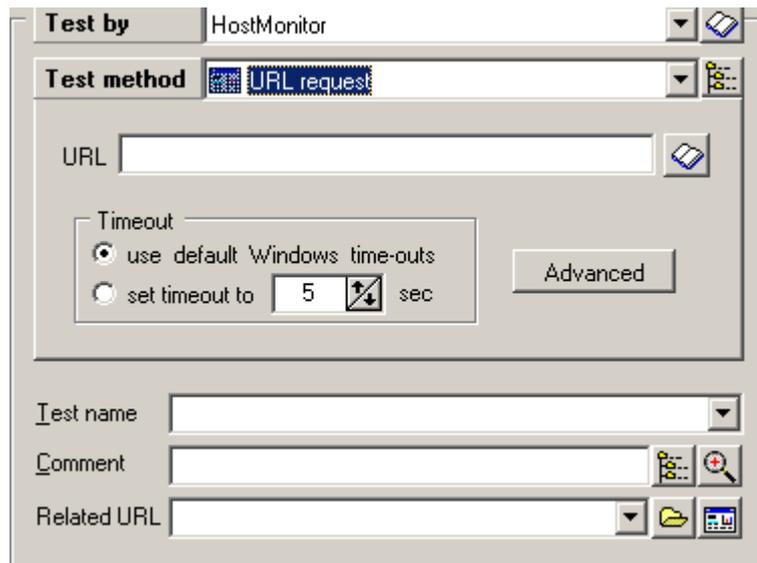
系统对外服务通常需要提供 TCP 端口，HostMonitor 提供了 TCP 端口存活状态的检测方法，类似于手动执行 Telnet 命令。使用此方法时，HostMonitor 会试图与指定主机的指定端口号建立 TCP 连接，如果超时则报警。也可以选择检测建立连接后服务器端返回的数据进行报警。

设置界面与 Ping 方法非常相似：



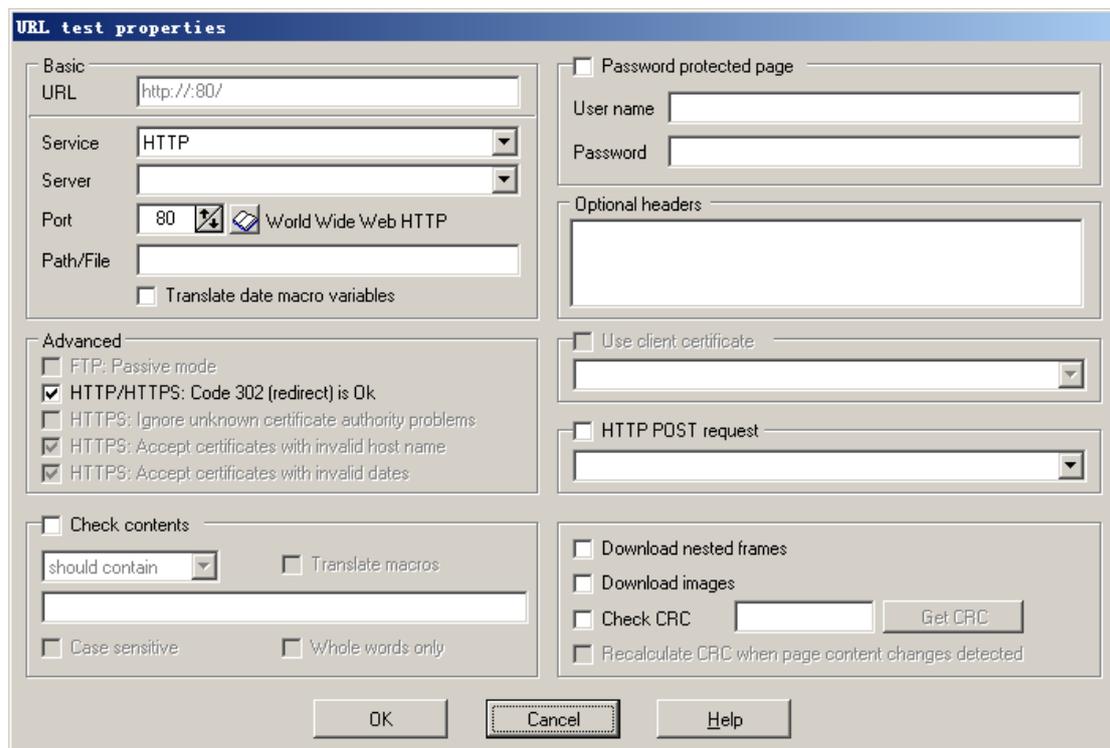
### 5.2 监控 HTTP、HTTPS、FTP 服务器

如果我们的系统建立了 Web 或 FTP 服务器，则可以使用 URL 请求（URL request）检测判断服务器是否正常工作，而不仅仅是判断端口是否开放。这样我们的监控手段也从传输层上升到了应用层，有能力直接监控业务运行状态。



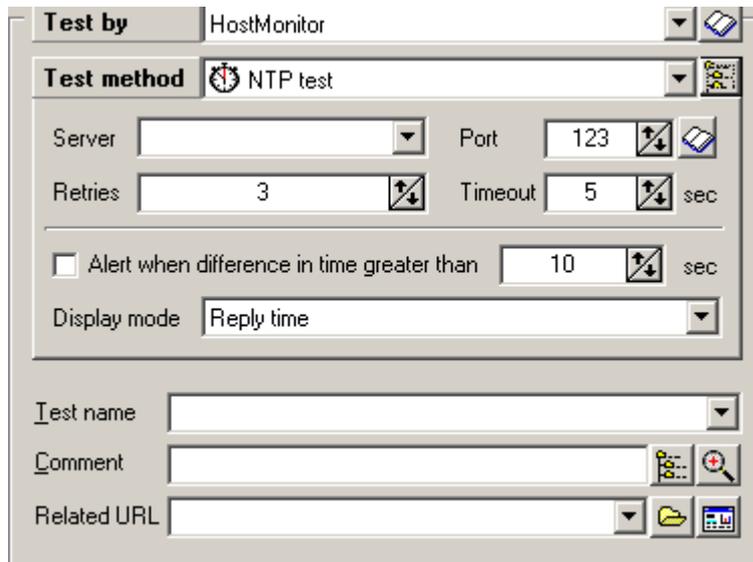
如果不需要特殊设置，直接填入 URL 即可，超时时间与 Windows 默认设置相同。

高级设置中有丰富的属性可以选择，例如服务类型（支持 FTP、HTTP、HTTPS）、端口号、是否支持重定向、是否忽略证书错误、是否检查返回内容、是否需要用户登录、配置请求头、配置 POST 参数、校验 CRC 等等：



## 5.3 监控 NTP 服务

很多情况下我们需要确保系统内的各服务器和网络设备的时间保持一致，因此需要用到网络校时协议（The Network Time Protocol，简称 NTP）。HostMonitor 同样提供了对 NTP 服务器存活状态的检测方法 NTP/SNTP：



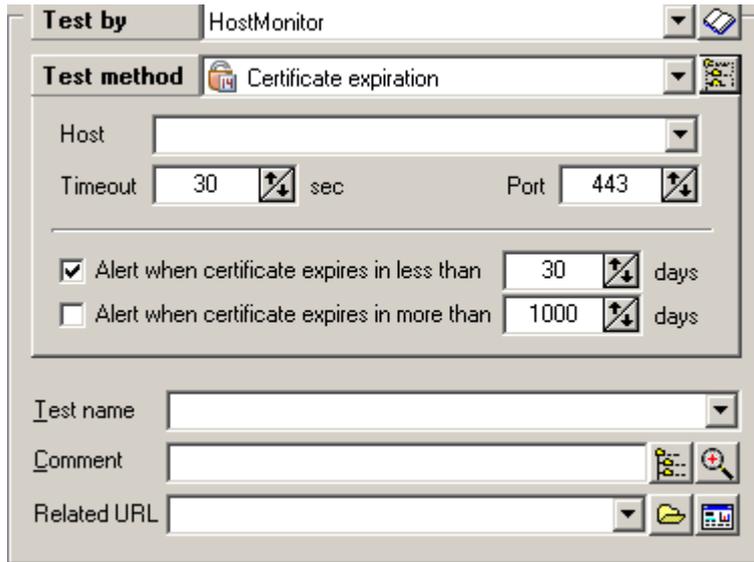
The image shows a configuration dialog for NTP testing in HostMonitor. The dialog is titled "Test by HostMonitor" and "Test method NTP test". It contains the following fields and controls:

- Test by:** HostMonitor
- Test method:** NTP test
- Server:** A dropdown menu.
- Port:** 123
- Retries:** 3
- Timeout:** 5 sec
- Alert when difference in time greater than 10 sec
- Display mode:** Reply time
- Test name:** A text input field.
- Comment:** A text input field.
- Related URL:** A dropdown menu.

设置方法非常简单，不再叙述。

## 5.4 监控 SSL 数字证书有效期

在部署了 HTTPS 的系统中，我们需要关注 SSL 证书的有效期，以便在到期前及时更换，防止过期。HostMonitor 提供了数字证书有效期的检测方法 Certificate check:



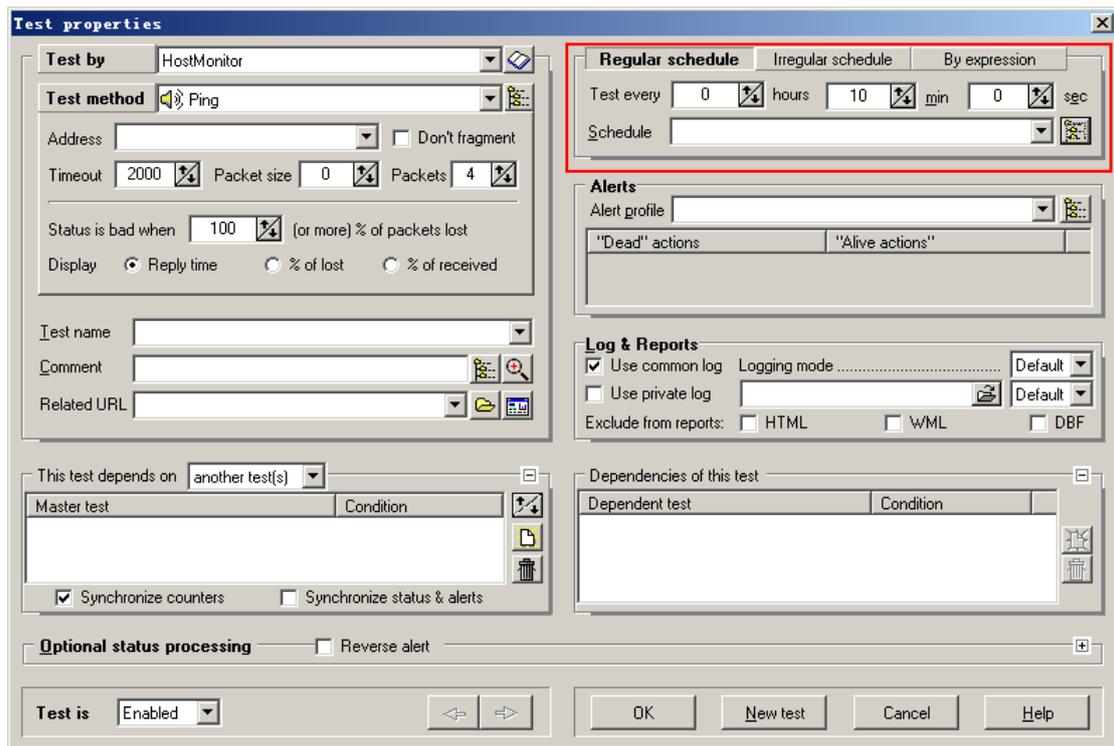
The image shows a configuration window for HostMonitor. The 'Test by' dropdown is set to 'HostMonitor'. The 'Test method' dropdown is set to 'Certificate expiration'. Below this, there are several input fields: 'Host' (empty), 'Timeout' (30 sec), and 'Port' (443). There are two checkboxes for alerting: 'Alert when certificate expires in less than 30 days' (checked) and 'Alert when certificate expires in more than 1000 days' (unchecked). At the bottom, there are fields for 'Test name', 'Comment', and 'Related URL', each with a dropdown arrow and some icons to the right.

可以选择有效期少于多少天的情况下报警。

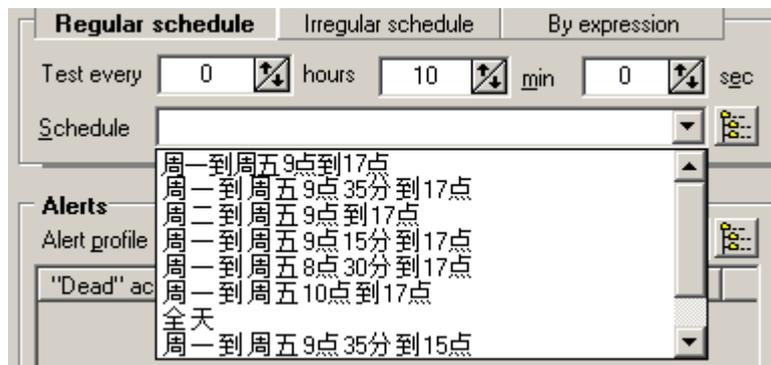
## 6 制定时间计划

监控方法默认会 24 小时不间断按照指定的时间间隔运行，有些监控我们希望只在特定的时间段执行，比如上班时间，又该怎么办呢？HostMonitor 提供了时间计划（Schedule）的功能。

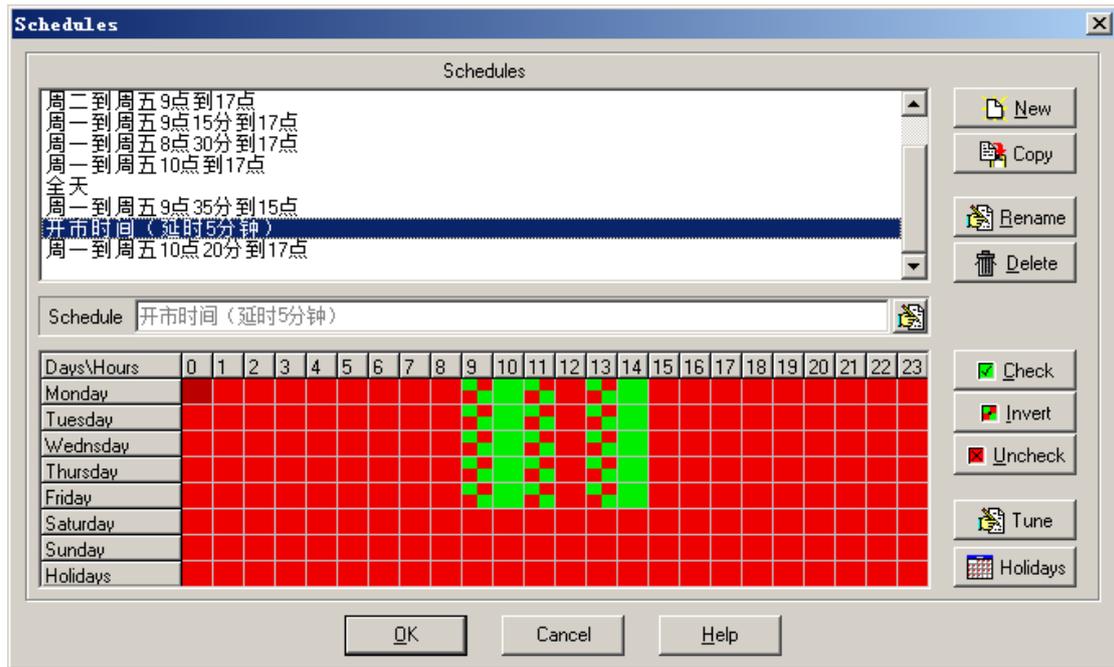
在检测方法设置页面的右上角是任务计划的设置，通常只要选择 Regular schedule 即可。Schedule 栏可以下拉选择已经创建好的时间计划。



HostMonitor 自带了一些时间计划，不过因为行业情况不同，未必符合我们的需要。我们还可以按照自身需求创建时间计划，例如我现在设置了这些：



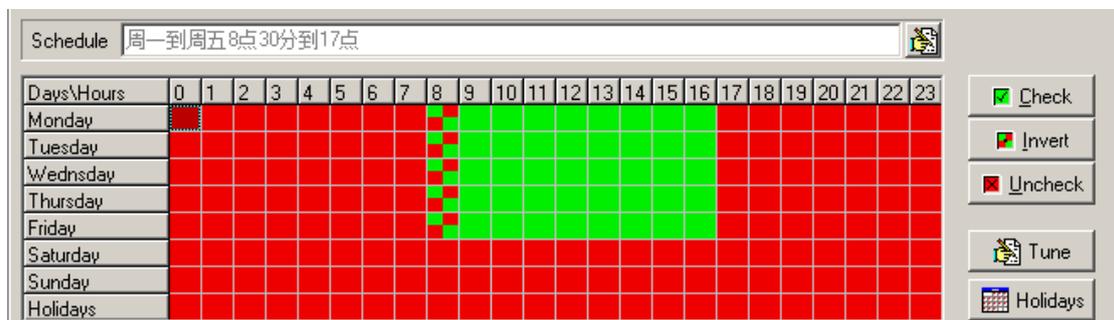
点击右侧按钮就会进入 Schedule 的管理界面：



其中图示部分的行标题是从周一到周日的的时间，最后一行为节假日，列标题为从 0 点到晚上 23 点的时间，这样每个方格都代表某一天的某个小时。红色部分表示该时间段内不执行监控（Uncheck），绿色部分表示该时间段内执行监控（Check），红绿相间的部分表示该时间段内只有一部分时间需要执行监控（Invert）。

创建自定义任务计划的大致流程是：

- 1、 点击 **New** 按钮创建新的时间计划并命名；
- 2、 新创建的时间计划是全天执行的，所有格子都是绿色。用拖拽的方式勾选不需要执行监控的时间，然后点 **Uncheck** 全部标记为不监控：



- 3、 点 **Tune** 按钮，进行分钟级别的修改：

Day	From	Till
周一	09:35	11:30
周一	13:05	14:59
周二	09:35	11:30
周二	13:05	14:59
周三	09:35	11:30
周三	13:05	14:59

这样就完成了时间计划的自定义，监控可以在我们需要的时间才执行了！

## 6.1 设置假期

时间计划的设置页面还可以设定一年中的假期，点击 **Holidays** 按钮弹出假期设定窗口，在窗口中选择需要设置为假期的日子，按空格键即可将当天标记为假日：

January							February							March							April									
周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日			
				1	2	3	1	2	3	4	5	6	7	1	2	3	4	5	6					1	2	3				
4	5	6	7	8	9	10	8	9	10	11	12	13	14	7	8	9	10	11	12	13	4	5	6	7	8	9	10			
11	12	13	14	15	16	17	15	16	17	18	19	20	21	14	15	16	17	18	19	20	11	12	13	14	15	16	17			
18	19	20	21	22	23	24	22	23	24	25	26	27	28	21	22	23	24	25	26	27	18	19	20	21	22	23	24			
25	26	27	28	29	30	31	29	28	29	30	31	25	26	27	28	29	30	25	26	27	28	29	30							
May							June							Jule							August									
周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日			
						1					1	2	3	4	5					1	2	3	1	2	3	4	5	6	7	
2	3	4	5	6	7	8	6	7	8	9	10	11	12	4	5	6	7	8	9	10	8	9	10	11	12	13	14			
9	10	11	12	13	14	15	13	14	15	16	17	18	19	11	12	13	14	15	16	17	15	16	17	18	19	20	21			
16	17	18	19	20	21	22	20	21	22	23	24	25	26	18	19	20	21	22	23	24	22	23	24	25	26	27	28			
23	24	25	26	27	28	29	27	28	29	30	25	26	27	28	29	30	31	29	30	31	29	30	31							
30	31																													
September							October							November							December									
周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日	周一	周二	周三	周四	周五	周六	周日			
				1	2	3	4						1	2					1	2	3	4					1	2	3	4
5	6	7	8	9	10	11	3	4	5	6	7	8	9	7	8	9	10	11	12	13	5	6	7	8	9	10	11			
12	13	14	15	16	17	18	10	11	12	13	14	15	16	14	15	16	17	18	19	20	12	13	14	15	16	17	18			
19	20	21	22	23	24	25	17	18	19	20	21	22	23	21	22	23	24	25	26	27	19	20	21	22	23	24	25			
26	27	28	29	30	24	25	26	27	28	29	30	28	29	30	26	27	28	29	30	31	26	27	28	29	30	31				
							31																							

Year 2016

结合时间间隔、时间计划和假期设置，我们几乎可以随心所欲控制每个监控的执行时间。

## 7 RMA 远程监控代理

深入使用 HostMonitor 就会发现远程监控有一些局限性，一些高级操作无法通过远程方式完成，例如监控远程服务器上的进程，或是需要跨网段监控，这种情况下就必须用到 HostMonitor 的远程监控代理程序 Remote Monitoring Agent，简写即为 RMA。

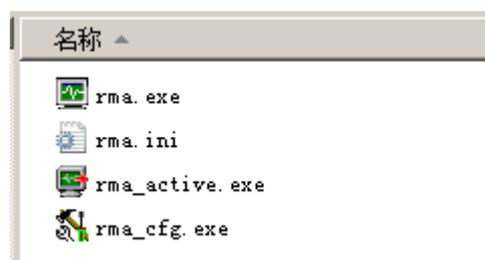
RMA 支持被动、主动两种工作方式，区别在于被动模式下 RMA 只会被动接受 HostMonitor 的连接，而主动模式下 RMA 会主动发起连接到配置指定的 HostMonitor 主机，通常使用被动模式即可。

RMA 支持 Win、Linux、BSD、AIX、Solaris 等平台，HostMonitor 在安装路径下自带了 Win 平台的 RMA，其他平台需要到网站上自行下载。

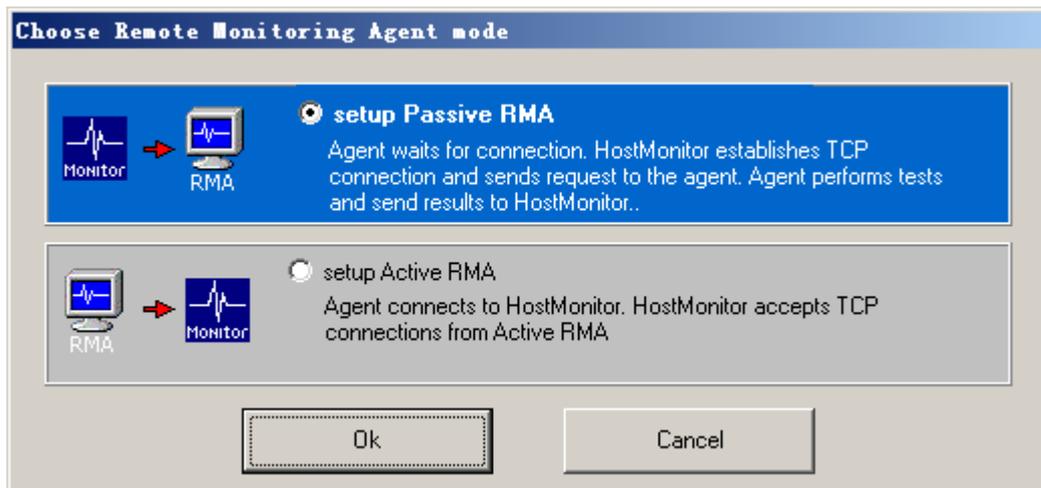
### 7.1 Windows 平台

HostMonitor 在安装路径下自带了 Win 平台的 RMA，放在 RMA-Win 目录中，共有四个文件：

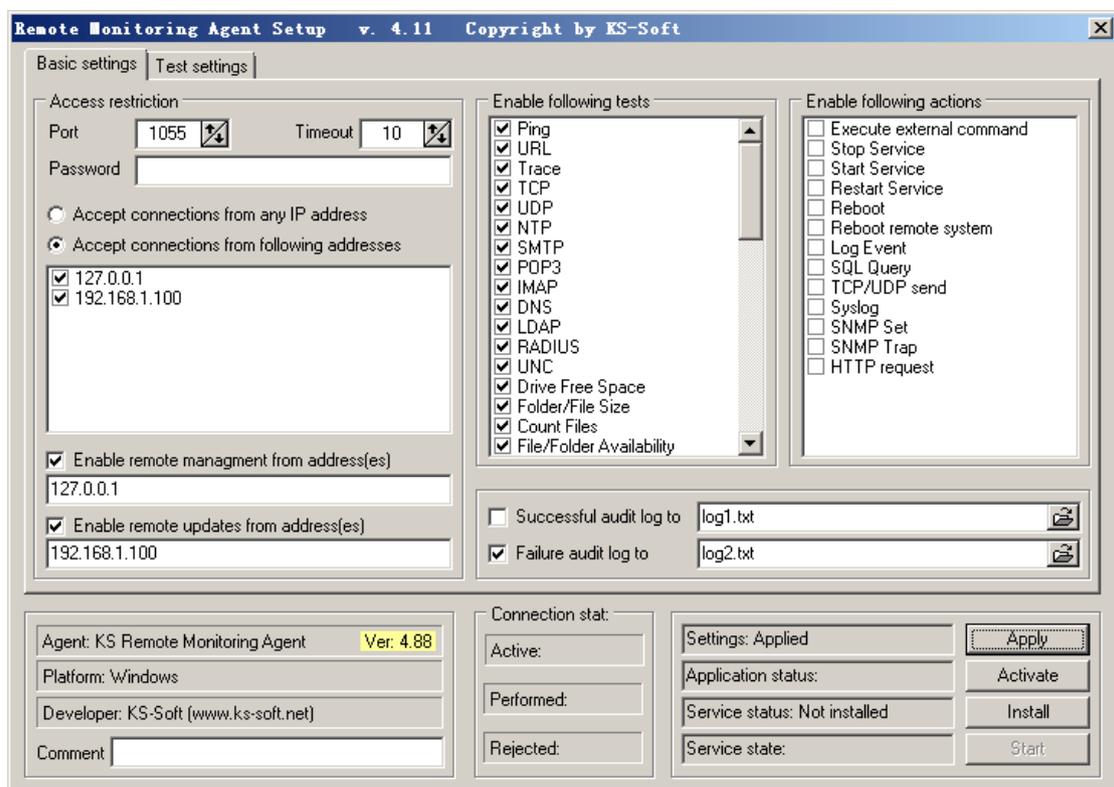
- rma.exe: 被动模式的 rma 程序
- rma.ini: RMA 配置文件
- rma\_active.exe: 主动模式的 rma 程序
- rma\_cfg.exe: RMA 图形化配置程序



使用时首先需要将 RMA-Win 目录整个复制到目标服务器中，运行 rma\_cfg.exe，会首先要求选择 RMA 的运行模式，默认选择被动模式 (setup Passive RMA) 即可。



然后就会打开详细配置页面：



从左上角开始，RMA 默认的端口号为 1055，一般不必修改。然后比较重要的是必须为 RMA 设置一个 Password，用来在连接时验证 HM 身份。对安全性要求较高的话，还可以设置允许哪些 IP 地址连接 RMA，双击修改即可。同时还可以设置允许远程管理或更新 RMA 的服务器 IP，便于日后的维护工作。

中间是允许 RMA 执行的测试类型，默认全部开启。

右侧比较重要，是允许 RMA 执行的操作类型，默认全部关闭，但根据实际需要，我们可以选择允许 RMA 执行外部程序（Execute external command）、重启系统服务（Restart Service）或干脆重启计算机（Reboot）。要注意根据自身需要

选择。

其他选项通常默认即可，现在就可以按下右下角的 **Apply** 按钮应用我们所做的修改了。除了 **Apply**，其他还有三个按钮：

**Activate:** 立即启动 RMA 程序

**Install:** 将 RMA 安装为系统服务，建议执行这个操作，保证 RMA 后台自动运行，也可以防止前台误操作关闭了 RMA 进程

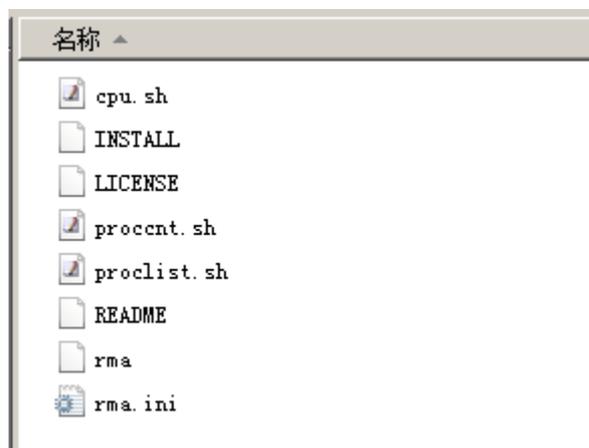
**Start:** RMA 安装为系统服务后，则可以用这个按钮开启或关闭它。

以上就完成了 Windows 版 RMA 的配置，可以在 HM 中使用它了！

## 7.2 Linux 及类 Unix 平台

除了 Windows 平台之外，其他平台的 RMA 需要自行到官网下载，这一节略过不表，只需注意根据需要下载 64 或 32 位版本即可。

下载下来的 RMA 同样有一个 `rma` 目录，里面文件较多，其中 `rma` 文件是主进程，我们关心配置文件 `rma.ini` 就好。



使用时可以先配置好配置文件再将 `rma` 目录复制到目标服务器中，也可以复制后再修改配置文件。注意复制后需要修改文件 `rma` 和 `.sh` 文件为可执行权限：

```
# chmod 755 rma
```

```
# chmod 755 *.sh
```

`rma.ini` 中的配置项和 Win 平台很类似，主要修改以下几项：

**RmaPath:** `rma` 程序文件的存放路径，注意是程序文件而不是目录的路径

**Password:** RMA 的连接密码，防止非授权的访问

**FilterActive:** 是否过滤来源 IP，默认 0 允许任意地址访问，可修改为 1 只允

许授权的 IP 地址访问

**FilterList:** 允许访问 RMA 的来源 IP 列表，空格键分隔，填 HM 主机地址

**UpdateAddr:** 允许更新 RMA 的 IP 列表，根据需要设置

**ManageAddr:** 允许远程管理 RMA 的 IP 列表，根据需要设置

其他参数请自行参照说明选择设置。

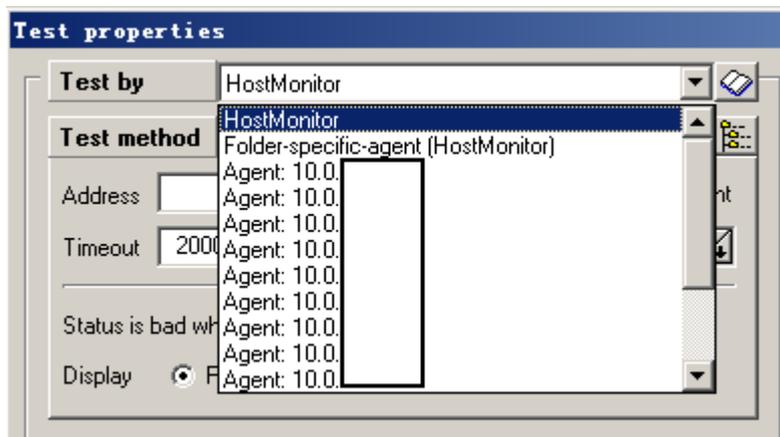
下一步我们来启动 rma 主进程，如果 RMA 程序和配置文件存放在 `/opt/rma` 目录下，则以后台方式启动 rma 进程的命令如下：

```
/opt/rma/rma -d /opt/rma/rma.ini
```

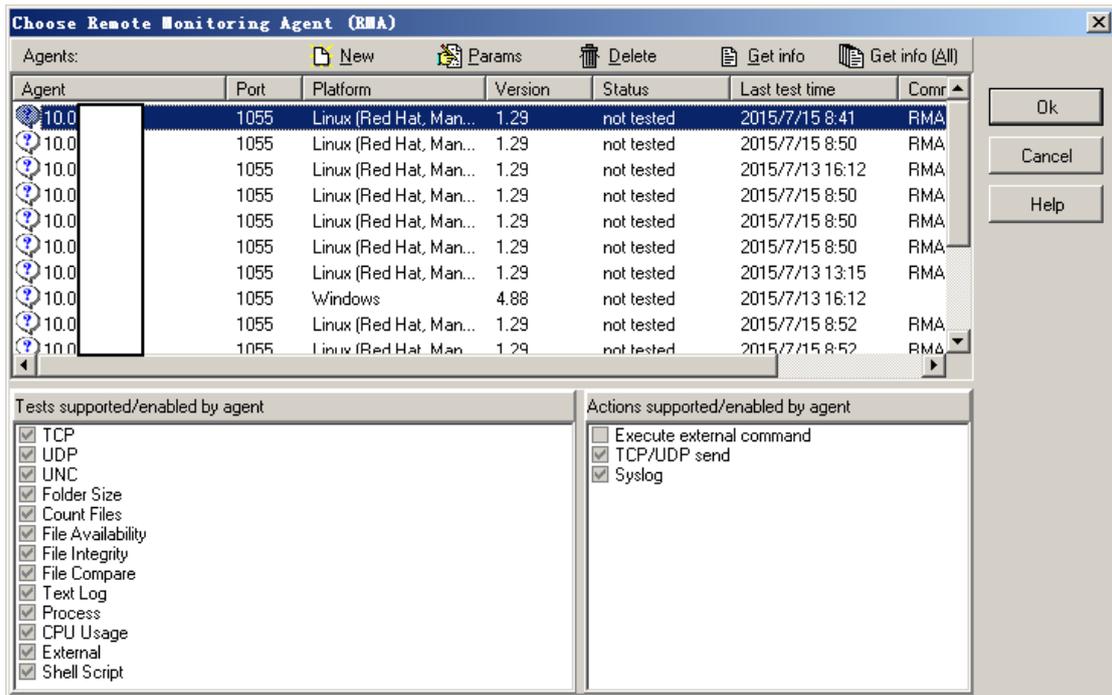
Red Hat Linux 系统中将以上命令加入 `/etc/rc.local` 即可实现 RMA 的开机自启动。

## 7.3 在 HostMonitor 中使用 RMA

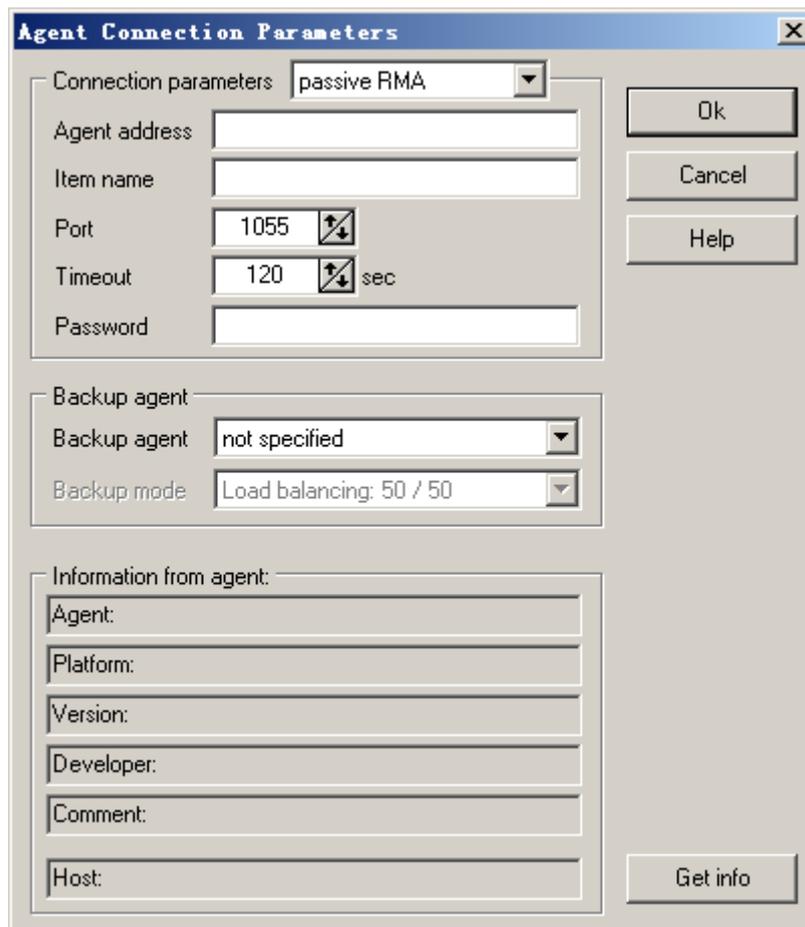
在 **Test** 的设置页面中，第一个选项就是选择监控程序 (**Test by**)，默认是 HM 本身，也可以是 RMA 程序：



点击右侧按钮打开 RMA 的管理界面：



点 New 按钮，添加一个新的 Agent:



主要需要填写的参数如下:

Agent address: RMA 的 IP 地址

Item name: 名称，自己能准确识别就好，一般也用 IP 地址

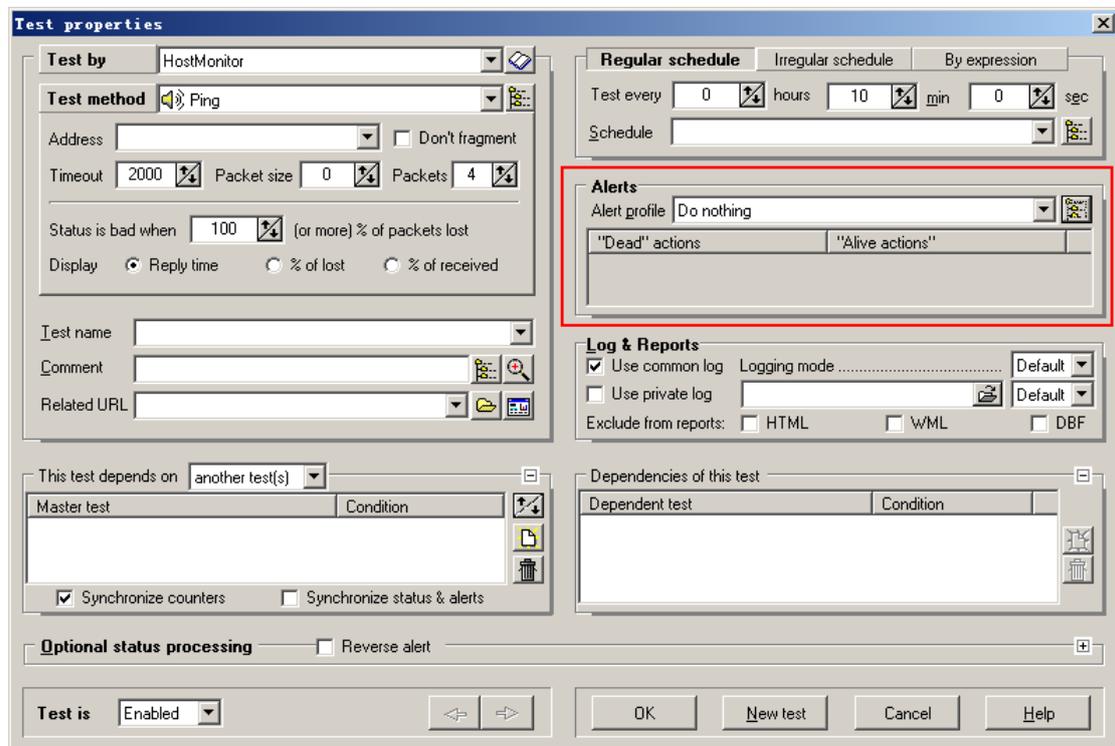
Password: RMA 的连接密码

其他参数一般默认即可。

这样添加后，在 Test 属性设置的 Test By 项中就可以选择这个 RMA 进行监控检测了。

## 8 个性化告警行为

HostMonitor 监控设置中的报警（Alerts）设置非常灵活，远不仅仅是报警这么简单，报警的同时还可以执行大量的动作（Action）。



HostMonitor 内置了几种报警配置文件做例子：

Do nothing: 什么也不做

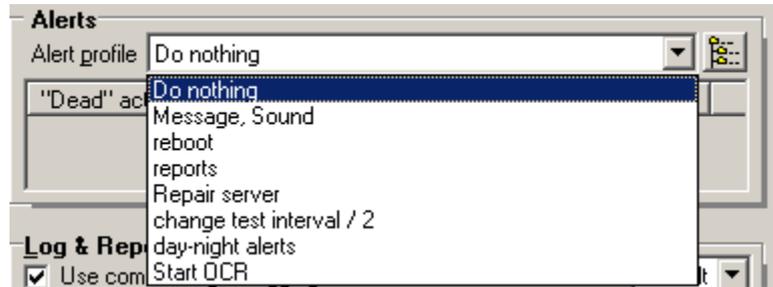
Message, Sound: 声光报警，包括 Show message, Play sound 两个动作

reboot: 重启远程服务器

reports: 生成报告文件

Repair server: 修复服务器，包括重启服务（Restart service）、发送数据到 TCP/UDP 端口（Send data to TCP/UDP port）两个动作。

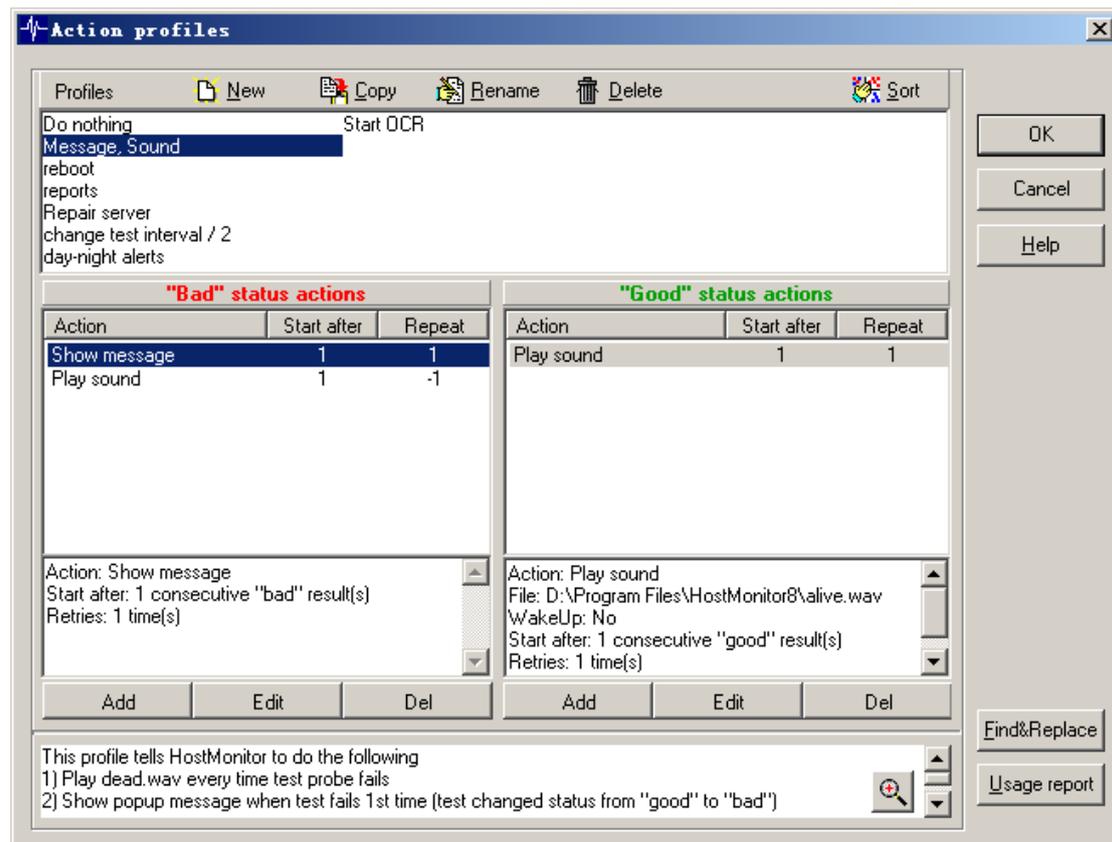
change test interval / 2: 修改测试的时间间隔



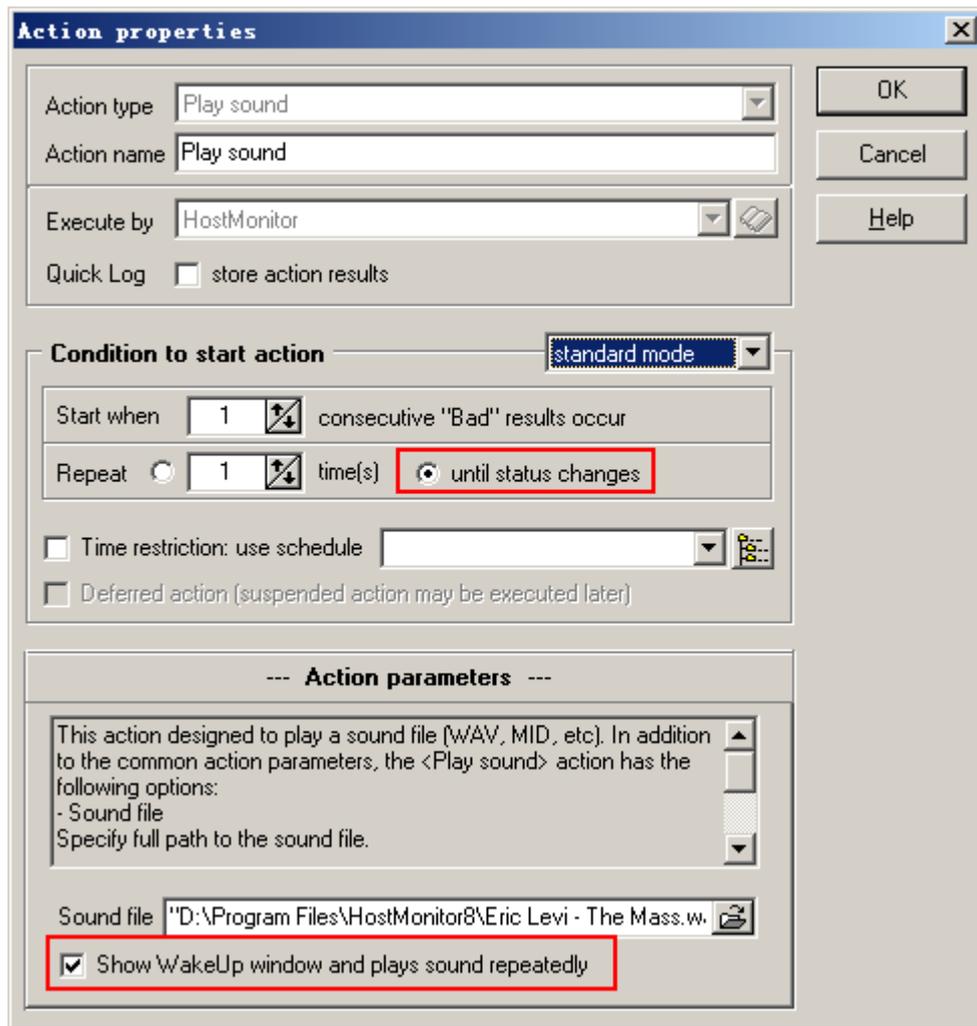
点击右侧按钮查看或修改各项报警行为。

## 8.1 声光报警

声光报警（Message, Sound）是最基本的报警方式，提供实施声光报警。打开它的设置页面：

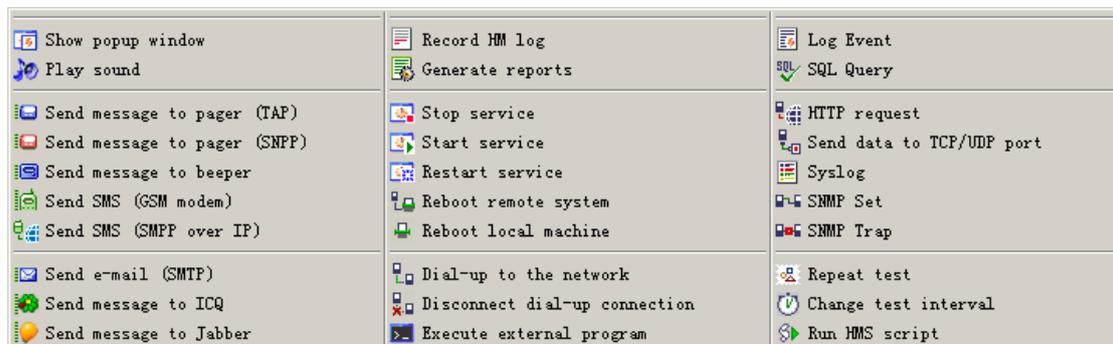


可以看到当监控转为 **Bad** 状态时，将会同时执行两个动作：**Show message** 显示消息弹窗和 **Play sound** 播放声音文件。双击可以修改具体的动作配置，例如我们可以将 **Play sound** 改为始终播放，同时显示 **WakeUp** 警报窗口。



## 8.2 执行动作

点击 Add 按钮可以为 Bad/Good 两种状态添加新的动作:



可以看出，允许执行的动作非常丰富，其中常用的包括：

Show popup window: 显示弹出窗口，即 Show message

**Play sound:** 播放声音文件，声音文件可以自定义，但必须为 wav 单声道格式

**Send e-mail:** 发送邮件

**Stop service:** 停止服务

**Start service:** 启动服务

**Restart service:** 重启服务

**Reboot remote system:** 重启远程服务器

**Reboot local machine:** 重启本机

**Execute external program:** 执行外部程序。可以用来自动启动程序，很有用

**SQL Query:** 执行 SQL 查询。非常有用

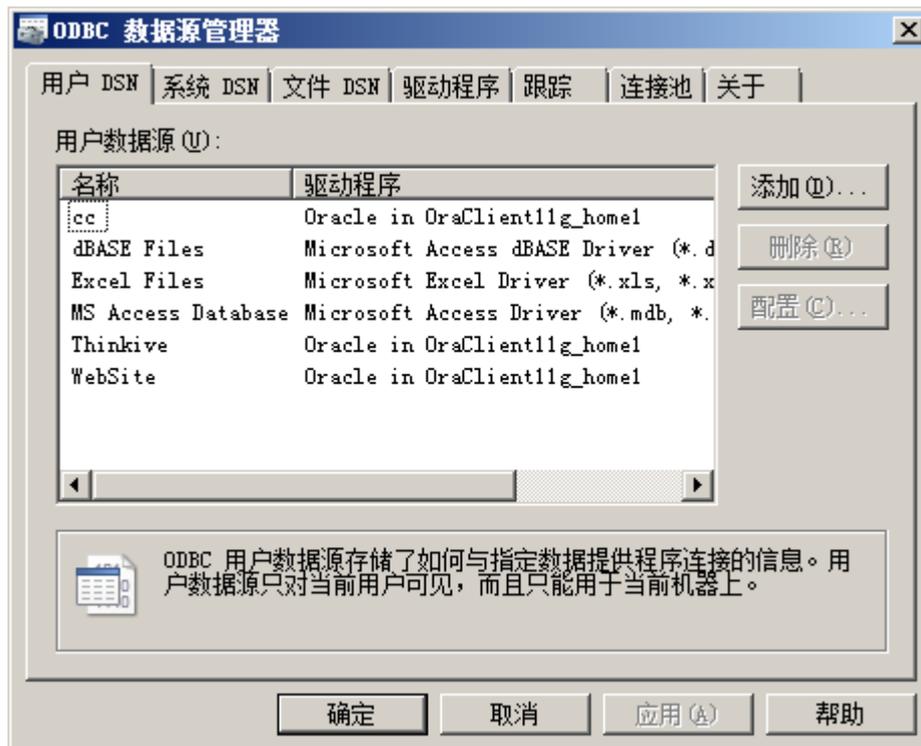
## 9 高级监控

结合 RMA、花样繁多的检测方法和告警动作，我们实际上可以让监控这件事玩得更加出彩，更加有声有色起来。

### 9.1 执行 SQL 语句并判断结果

大部分系统运行中会涉及数据库处理操作，例如客户数据采集、行情转换等等。很多情况下数据库处理结果非常重要，需要对其进行监控，以及时发现问题，HostMonitor 提供了非常灵活的方法允许执行 SQL 语句，并在特定条件下报警。下面以 Oracle 数据库为例进行说明：

HostMonitor 提供的是 ODBC 查询数据库检测方法（ODBC Query），因此首先需要在监控机上建立数据库的 ODBC 连接，需要注意由于 HostMonitor 是 32 位的软件，因此在 64 位操作系统下必须通过 32 位的 ODBC 管理程序“odbcad32.exe”来创建 ODBC 连接，这个程序 Win7 系统中位于 C:\Windows\SysWOW64 文件夹中。32 位系统中则可以直接使用控制面板管理工具里的 ODBC 数据源打开。



选择添加，如果安装过 Oracle Client，就能看到 Oracle 的 ODBC 驱动程序：



然后弹出 Oracle ODBC 驱动配置窗口，其中各项参数包括：

Data Source Name: ODBC 数据源名称，重要，后面需要在 HostMonitor 中使用

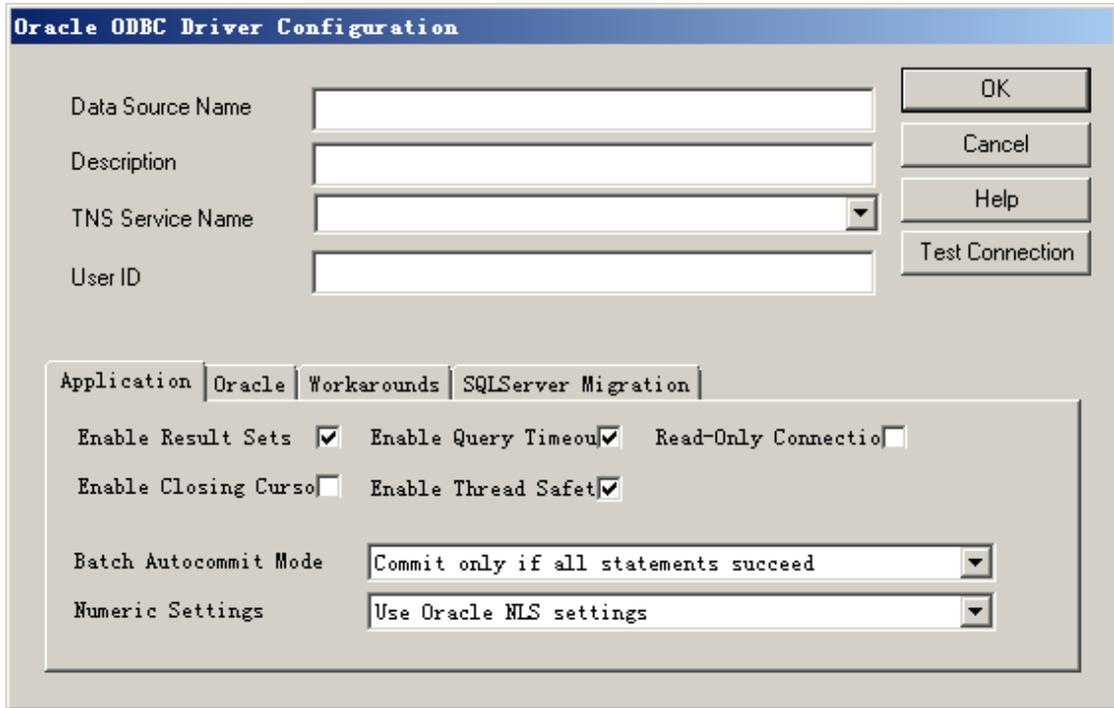
Description: 描述，可以不填

TNS Service Name: tnsnames.ora 文件中已经配置的 TNS 服务名称，Oracle

数据库管理员应当都知道是什么含义，直接下拉选择即可。

**User ID:** 用于登录的数据库用户名

其它选项一般无需修改，设置完成后可以点击 **Test Connection** 测试数据库连接是否已经正确配置。



The image shows the 'Oracle ODBC Driver Configuration' dialog box. It has a title bar with the text 'Oracle ODBC Driver Configuration'. Below the title bar, there are four input fields: 'Data Source Name', 'Description', 'TNS Service Name' (with a dropdown arrow), and 'User ID'. To the right of these fields are four buttons: 'OK', 'Cancel', 'Help', and 'Test Connection'. Below the input fields, there is a tabbed interface with four tabs: 'Application', 'Oracle', 'Workarounds', and 'SQLServer Migration'. The 'Oracle' tab is selected. Under the 'Oracle' tab, there are several settings: 'Enable Result Sets' (checked), 'Enable Query Timeou' (checked), 'Read-Only Connectio' (unchecked), 'Enable Closing Curso' (unchecked), and 'Enable Thread Safet' (checked). Below these are two dropdown menus: 'Batch Autocommit Mode' (set to 'Commit only if all statements succeed') and 'Numeric Settings' (set to 'Use Oracle NLS settings').

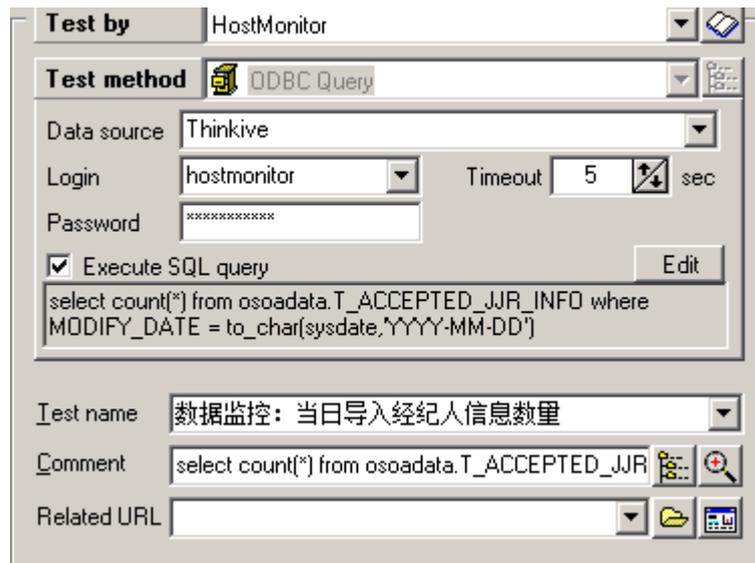
配好了数据源，下面就可以回到 HostMonitor 中，添加 ODBC Query 检测方法了，对应的参数说明如下：

**Data source:** 也就是上面创建的 ODBC 数据源的 Data Source Name

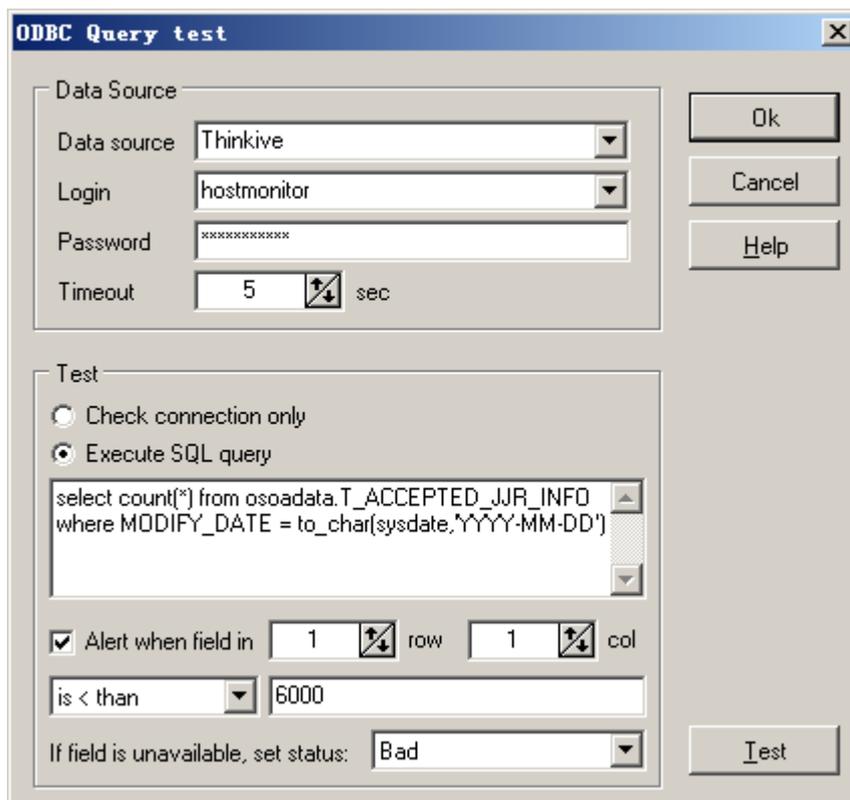
**Login:** 数据库登录用户名，注意要有相关的查询权限

**Password:** 密码

**Timeout:** 超时时间，默认 5 秒，一般无需修改



这里最重要的是 **Execute SQL query** 即是否执行 SQL 查询。如果不勾选，则只检测数据库连接状态，我们当然要选中的！然后点 **Edit** 就可以输入 SQL 语句了：



上方依然是数据源信息，下方文本框中粘贴 SQL 语句即可，**注意不要用分号“;”结尾，会报错的**。考虑到这是最简陋的文本框，建议大家还是在其他 SQL 工具中测试好 SQL 语句后直接粘贴进来。再往下的信息很好理解，当返回值的第 1 行第 1 列小于 6000 时即报警，如果字段不可用也报警。这里就按照自己的实际需要灵活设置。

以上就完成了 SQL 查询检测的配置，结合任务计划，我们可以让 SQL 查询在正确的时间执行，例如每个工作日的早晨。这样 HostMonitor 就会成为我们得力的工作助手，自动化帮我们处理原本需要人工定时检测的工作了！我们运维人员就可以解放出来喝喝早茶……哦不对，从事更需要创造力的工作:-)

### 9.1.1 监控 Oracle 数据库会话数

Oracle 数据库会话数是非常重要的一项指标，一旦超出就会导致业务系统无法创建新连接，甚至危及系统运行。在 Oracle 数据库中，当前会话数可以用这条 SQL 语句查询：

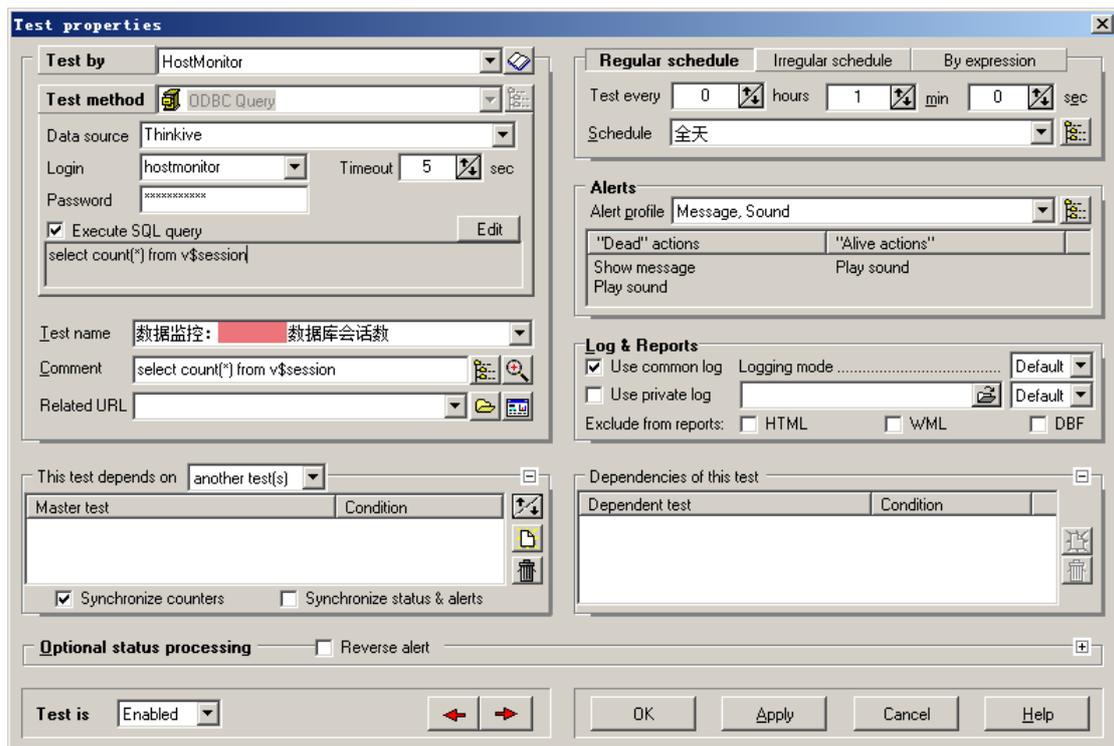
```
select count(*) from v$session
```

所以，利用 ODBC Query 功能和这条 SQL 语句，我们完全可以实时监测 Oracle 的会话数量，提前解决故障。下面我们来看一看如何实现这个目的。

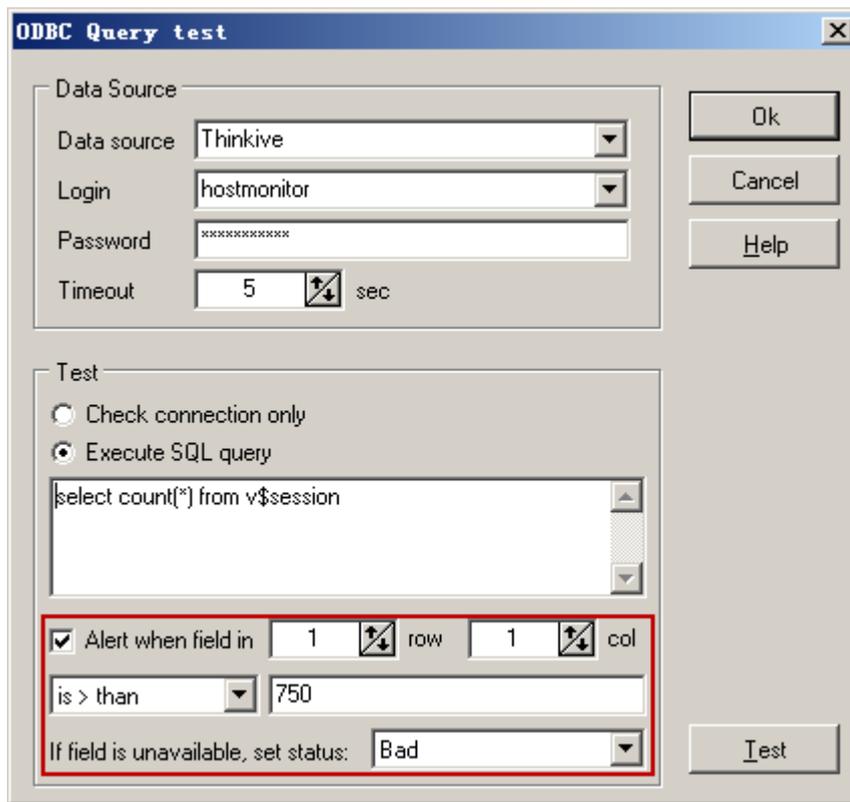
为了安全起见，第一步我们要做的是在 Oracle 数据库中建立用于 HostMonitor 监控的用户，并授予必要的 CONNECT 角色（授权连接数据库）和 SELECT ANY DICTIONARY 权限（授权查询 v\$session 数据表）。

第二步，按照前文的说明，创建 ODBC 数据源。

第三步，在 HostMonitor 中创建 ODBC Query 监控任务。



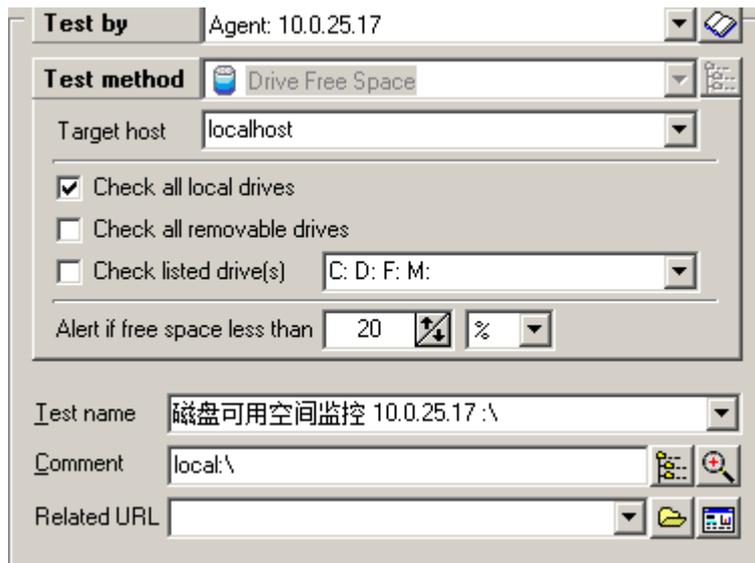
查询参数设置为会话数量超过 750 则报警，这里根据实际需要调整参数值：



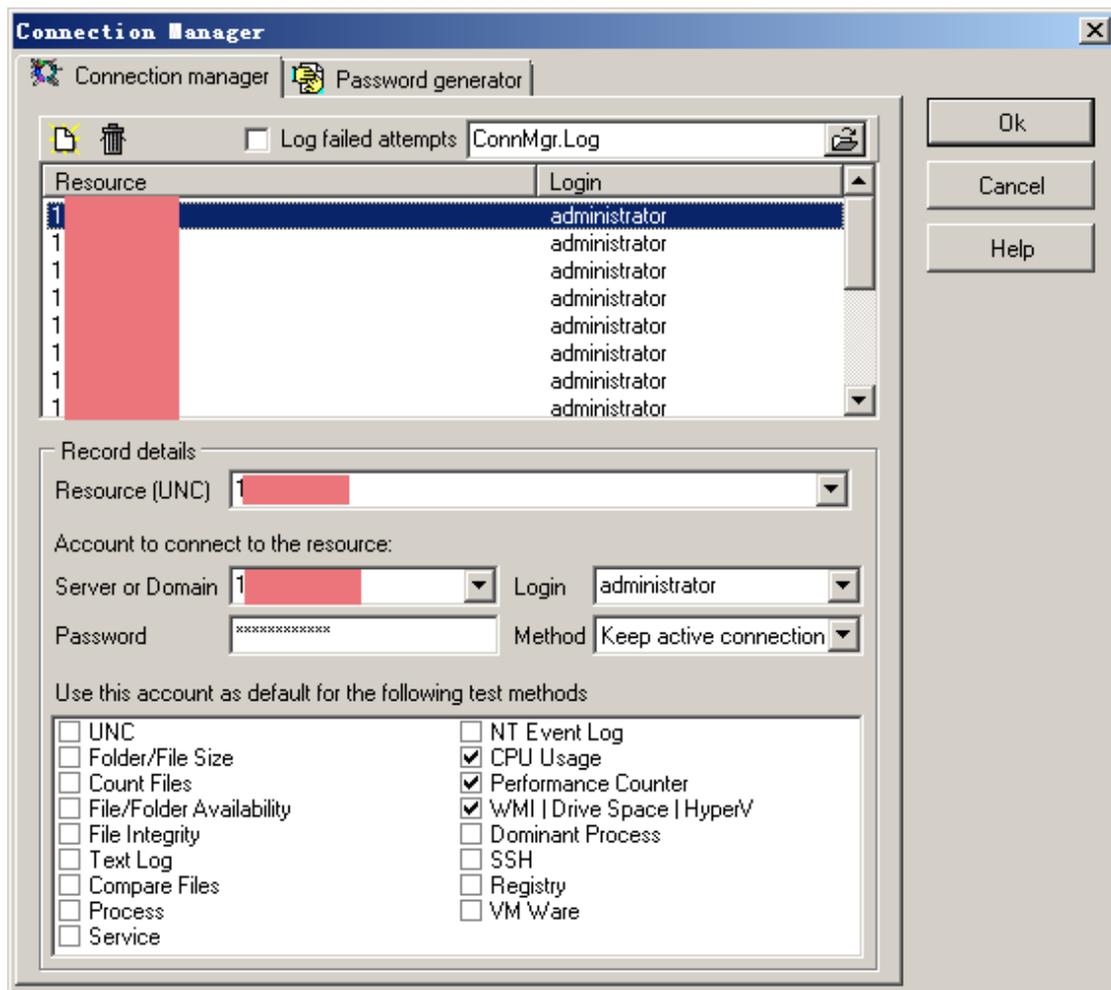
通过这样的设置，HostMonitor 就能帮助系统管理员实时监控数据库会话数状态，在超出正常范围后立即报警，这样，系统管理员就会有充分的时间排查系统问题，找出哪个服务器发起了最多的连接，并在问题影响到系统正常运行之前就予以解决。

## 9.2 Windows 磁盘可用空间

HostMonitor 自带了 Drive Free Space 检测方法，但由于依赖 WMI 协议，因此仅能用于 Windows 平台。可以使用 RMA 或者远程连接两种方式，下图以 RMA 连接方式为例，磁盘空间小于 20% 时即报警，非常简单：



如果服务器没有安装 RMA，也可以使用远程方式进行监控，前提是服务器没有禁用 WMI。首先要配置远程服务器的连接账户，打开 Profiles 菜单中的 Connection Manager 菜单窗口：



这里可以添加远程服务器的连接方式，以及使用此账户允许执行的监控操作，

需要启用 WIM|Drive Space|HyperV 检测方法。主要参数有：

Resource(UNC)：资源名称，服务器 IP 即可

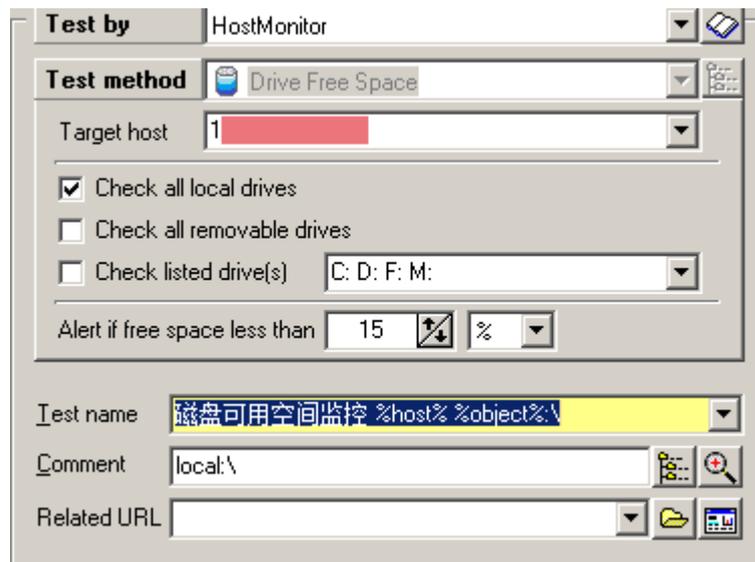
Server or Domain：服务器 IP 或域名

Login：登录用户

Password：登录密码

Method：可选始终保持连接，或按需连接

然后添加 Test，Target host 选择刚刚增加的服务器即可：



可以选择按照可用空间的百分比或绝对大小报警。添加后在列表中即可看到详细信息，其中 Reply 字段显示的是服务器当前的最小可用空间。

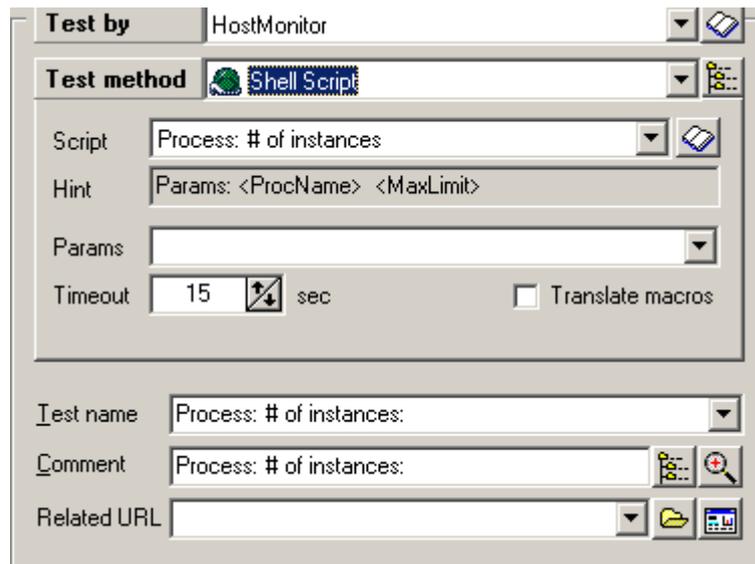
Test name	Status	Recurr...	Reply	Test m
Root\Drive Free Space\				
磁盘可用空间监控 1	Disabled	4	8 %	Drive s
磁盘可用空间监控 1	Disabled	6	Error: 拒绝访问。	Drive s
磁盘可用空间监控 1	Disabled	2	4 %	Drive s
磁盘可用空间监控 1	Ok	36	45 Gb	Drive s
磁盘可用空间监控 1	Ok	110	46 %	Drive s
磁盘可用空间监控 1	Ok	65	39 %	Drive s
磁盘可用空间监控 1	Ok	110	30 %	Drive s
磁盘可用空间监控 1	Ok	110	44 %	Drive s
磁盘可用空间监控 1	Ok	110	63 %	Drive s
磁盘可用空间监控 1	Ok	110	71 %	Drive s
磁盘可用空间监控 1	Ok	110	68 %	Drive s
磁盘可用空间监控 1	Ok	110	68 %	Drive s

## 9.3 执行 Shell 脚本并判断结果

对于 Linux 服务器, HostMonitor 很强大的一项功能是可以执行用户自定义的 Shell 脚本并做条件判断报警, 这样我们实际上可以对系统实现非常灵活的监控。

Shell 脚本必须通过 RMA 本地执行, 不能远程运行。

Test method 选择 Shell Script, 即可看到如下界面:



主要参数包括:

**Script:** 选择要使用的脚本名称。右侧按钮可以打开脚本管理界面

**Hint:** 脚本使用提示。例如“Params:<ProcName> <MaxLimit>”表示此脚本需要输入两个参数, 第一个是进程名称, 第二个是最大数量, 中间以空格分隔。

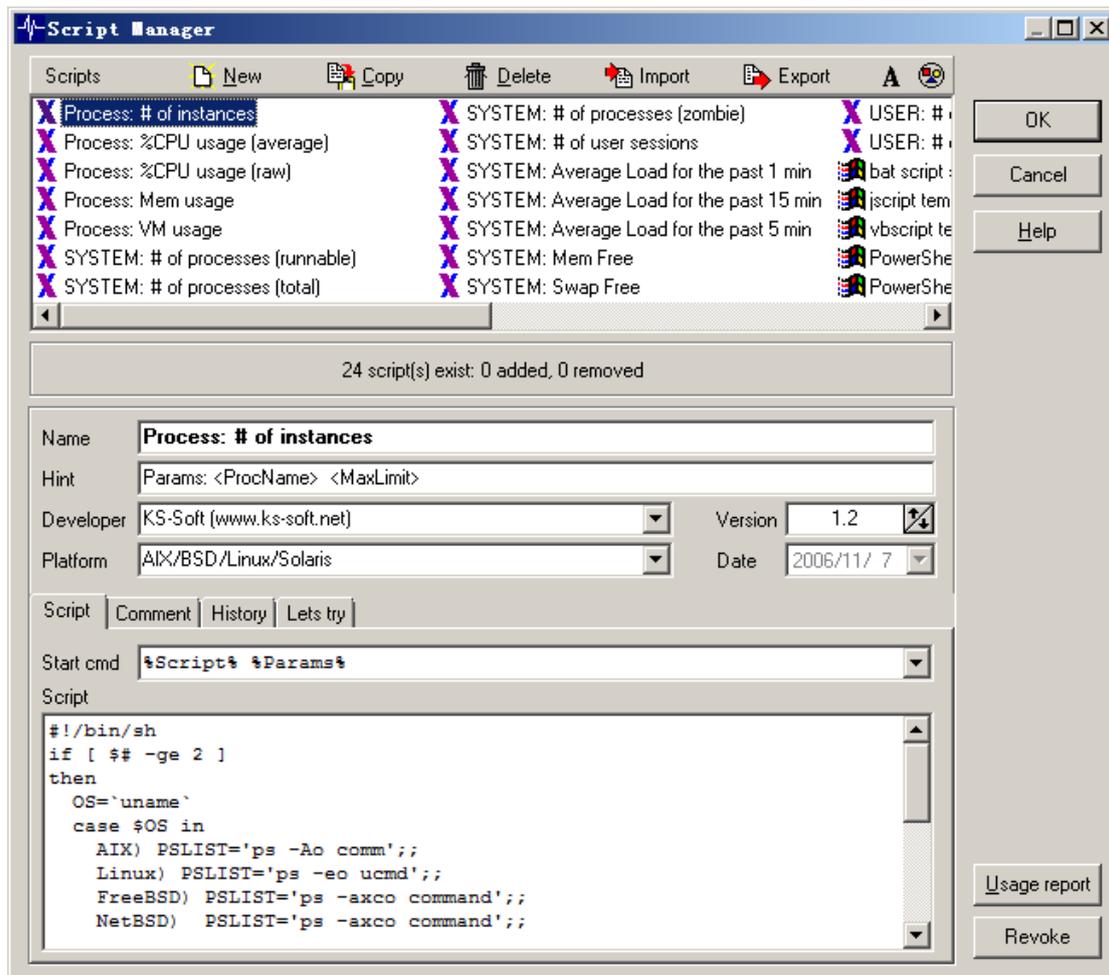
**Params:** 输入脚本参数。按照上方提示输入即可

HostMonitor 本身自带了一些常用脚本, 支持多平台使用。例如磁盘使用率、进程数量、交换分区可用空间等等。同时, 还可以按照实际需要, 由我们自定义 Shell 脚本, 下面我们就来尝试一下。

### 9.3.1 监控 Linux 磁盘可用空间

HostMonitor 没有提供 Linux 系统专用的的磁盘可用空间检测方法, 但是有了超级灵活的自定义 Shell 脚本工具, 我们完全可以自己实现 Linux 平台下的磁盘可用空间监控。

打开脚本管理器 (Script Manager), 我们看到如下图的界面:



可以看到许多内置的 Shell 脚本，下方则是 Shell 脚本的具体内容，其中主要参数有：

**Name:** Shell 脚本名称

**Hint:** 使用提示

**Developer:** 脚本开发者

**Version:** 版本号

**Platform:** 此脚本支持哪些平台

**Script:** 脚本内容，其中 **Start cmd** 表示脚本的使用方法，不同系统平台中有所区别

**Comment:** 注释

**History:** 版本记录

**Lets Try:** 脚本测试工具

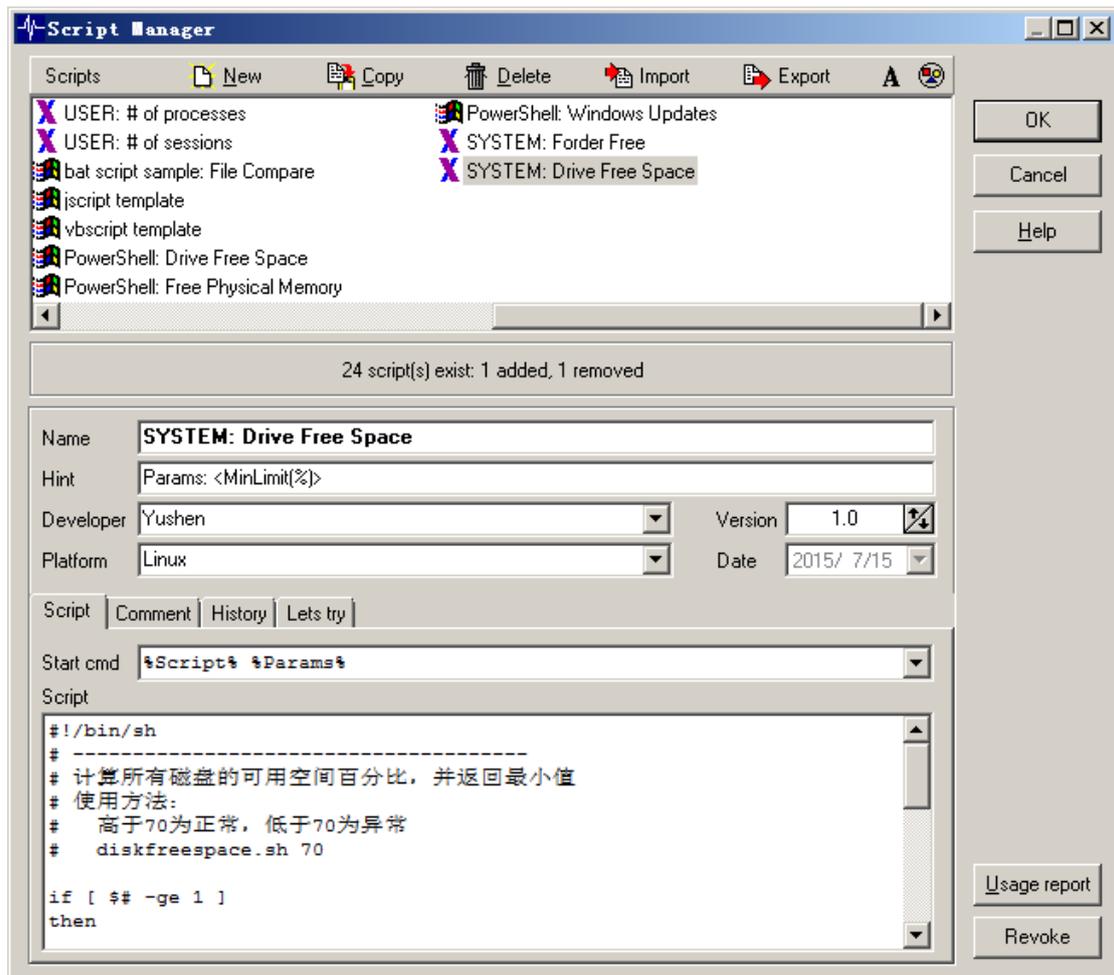
现在,就让我们来创建一个自己的脚本吧。点击 **New** 按钮,Name 填“SYSTEM: Drive Free Space”, Hint 填“Params: <MinLimit(%)>”, Developer 填上自己的大名, Platform 填 Linux, Version 设置为“1.0”, Start cmd 填“%Script% %Params%”, 然后把下面的脚本内容完整复制到 Script 栏中:

```
#!/bin/sh
# -----
# 计算所有磁盘的可用空间百分比,并返回最小值
# 使用方法:
# 高于 70 为正常,低于 70 为异常
# diskfreespace.sh 70
if [ $# -ge 1 ]
then
    OS=`uname`
    case $OS in
        Linux) df -hP | awk '{if(NR > 1){ print(100-$5,$6)}}' | sort
        -gr | tail -1 | awk '{if ($1 >= '$1') {printf("ScriptRes:Ok: %s
        剩余 %d %\n",$2,$1)} else {printf("ScriptRes:Bad:%s 剩余 %d %\n",
        $2, $1)} }';;
        *) echo 'ScriptRes:Unknown:script is not designed for '$OS
        exit;;
    esac
else
    echo 'ScriptRes:Unknown:缺少足够的参数'
fi
# 自定义 Shell 脚本到此结束
```

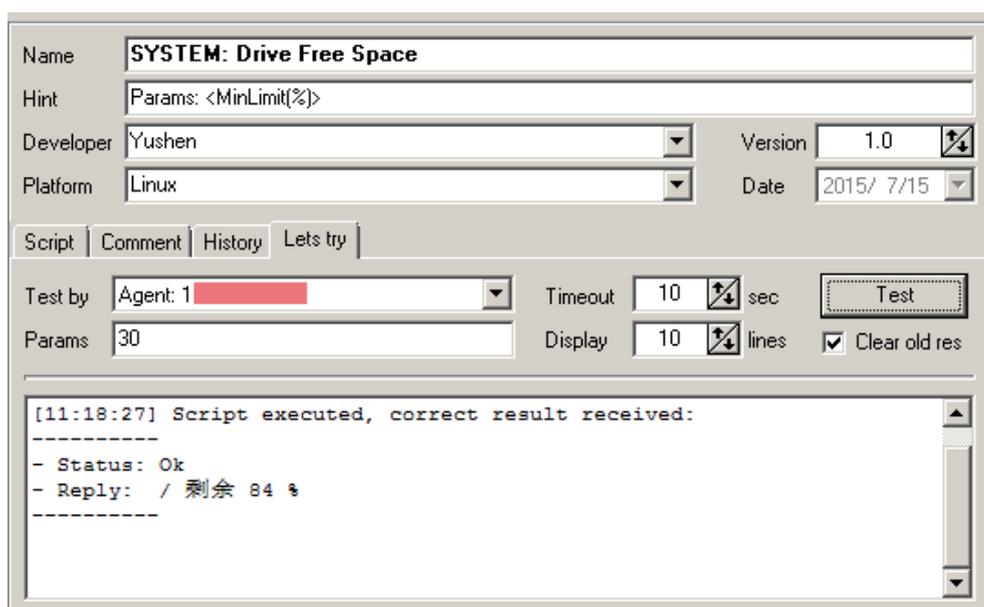
这个脚本会返回所有分区的可用空间比例中最小的值,一般来说足够用了。

Shell 脚本的编写方法不在本文讨论范围内,有兴趣的朋友可以自行研究。

填写后,效果如图所示:



然后点开 Lets try，测试下脚本是否能够正常工作。Test by 选择一台已经安装了 RMA 并开启了 Shell Script 检测方法的 Linux 主机，Params 填写上最小可用空间百分比，点击 Test，即可看到执行结果。

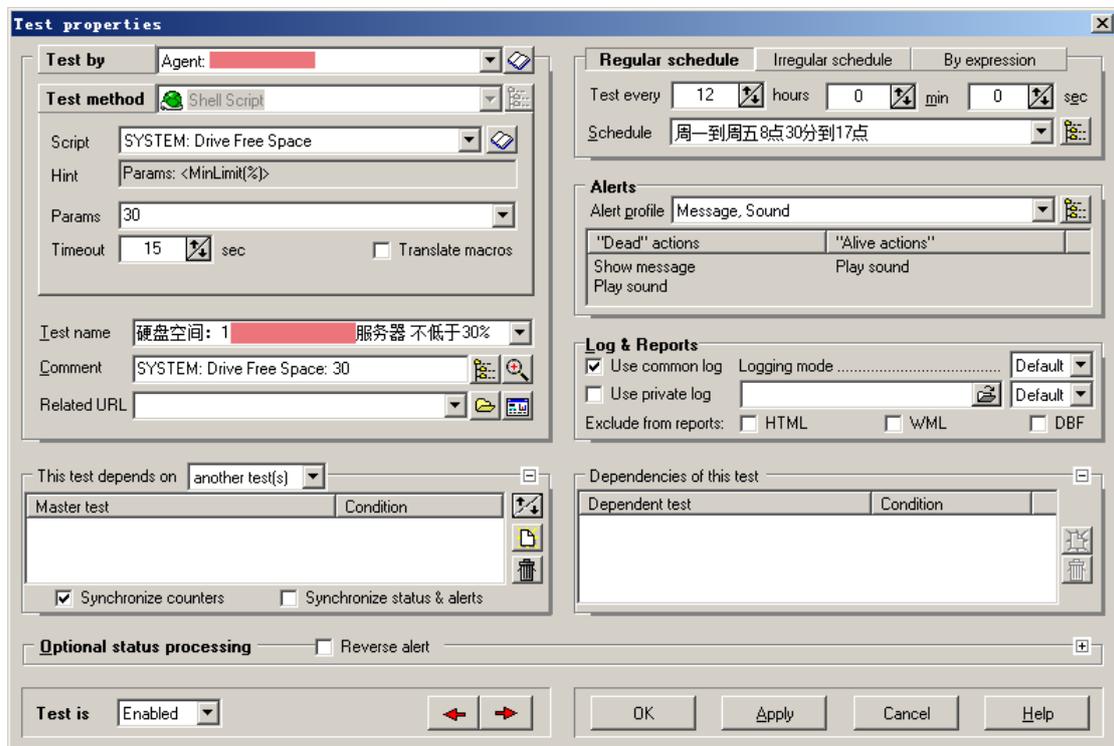


这里需要和实际情况做一下比较，检查下脚本执行结果是否正确。我们看一下这台主机的实际分区使用情况：

```
[root@10 ~]# df -h
文件系统          容量  已用  可用  已用%% 挂载点
/dev/sda2          101G  15G   82G   16% /
tmpfs              16G   0    16G   0% /dev/shm
/dev/sda1          200M  260K  200M   1% /boot/efi
/dev/sda4          410G  731M  388G   1% /home
```

剩余空间最小的是/dev/sda2，实际剩余 84%，完全符合预期。

然后就可以添加检测方法了，如下图所示（记得要使用 RMA），Script 选择我们刚刚添加的脚本名称“SYSTEM: Drive Free Space”，Params 填最小可用空间百分比：

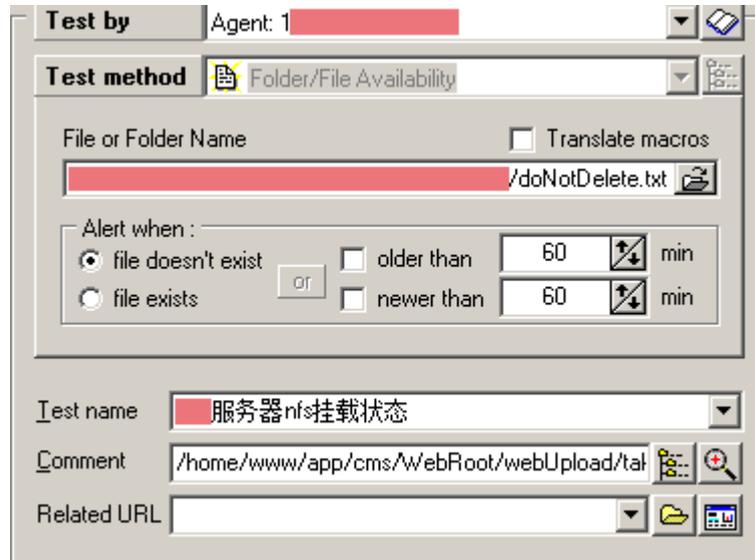


这样我们就创建了一个基于自定义脚本实现的 Linux 磁盘可用空间的监控，磁盘空间低于警戒线时即自动报警。由于脚本只需要创建一次，因此随后添加其他服务器的监控会非常方便快速。

## 9.4 监控 NFS 挂载状态

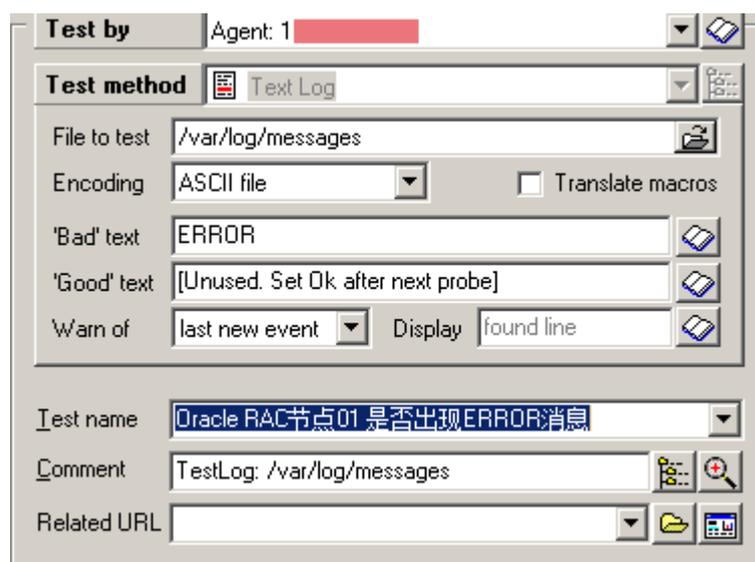
一些业务系统中需要用到 NFS 实现文件共享，有时候 NFS 未能正确挂载或挂载失败了，会导致业务上的异常，所以需要对 NFS 挂载状态进行实时监控。HostMonitor 没有直接提供 NFS 状态检测，但提供了文件可用性（Folder/File

Availability) 检测方法，同样可以实现我们的目的。办法很简单，在 NFS 中建立一个专用文件，例如 doNotDelete.txt，然后在目标服务器中检测这个文件的可用性即可确保 NFS 正常工作。如下图所示：



## 9.5 监控日志文件内容

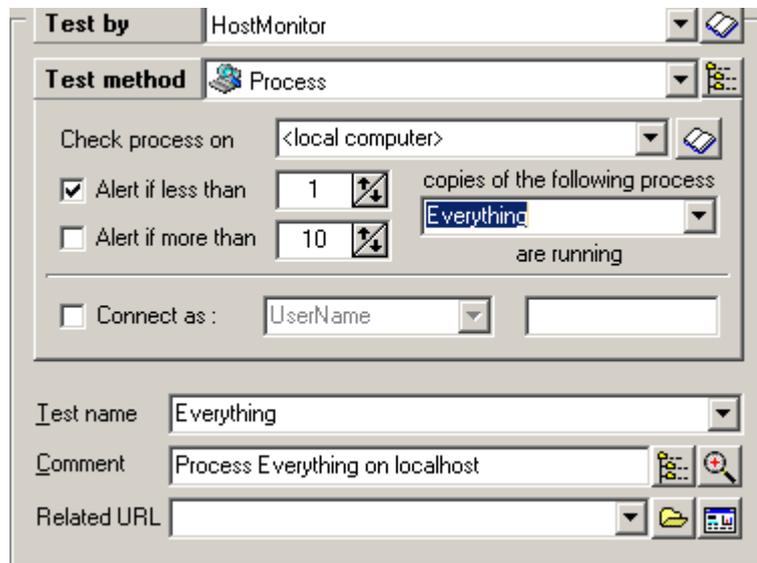
HostMonitor 可以使用关键字监控日志文件，符合条件时自动报警。这个功能相当有用，比如当 Oracle 数据库出现 Error 信息时，我们能够第一时间获得消息并采取行动，避免数据库宕机，我们可以这样设置：



注意实际中由于只能监控本地日志文件，因此必须使用 RMA。

## 9.6 监控进程数量

系统运维过程中，有时候会遇到程序不稳定自动退出的情况，需要监控进程个数，并在异常的时候报警。这时候可以使用 Process 检测方法：



上图中表示监控本机运行的名为“Everything”的进程，当少于 1 个时即报警。Process 可以监控本地或局域网中的服务器上的进程，推荐使用 RMA 进行本地监控。

### 9.6.1 程序退出后自动重启

最棒的是，利用报警动作执行外部程序（Execute External Program），我们还可以让 HostMonitor 自动启动异常退出的程序。

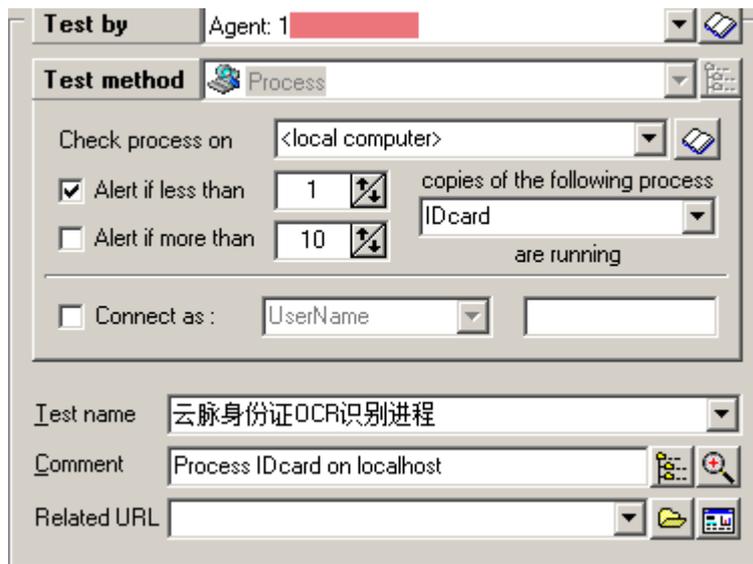
公司部署了一套厦门云脉的身份证 OCR 程序，这是一个 Windows 平台的命令程序，运行时产生 1 个进程“IDcard.exe”，长时间运行时偶尔会自动退出，而业务要求这个程序需要长期稳定运行，我们就以此程序为例。

经过测试发现 HostMonitor 在执行外部程序动作中直接远程启动命令程序存在一些问题，在远程桌面中不能正常显示程序窗口，不很方便，因此第一步需要在远程服务器上将要执行的命令写成 bat 批处理文件，例如 D:\StartOCR.bat，内容为：

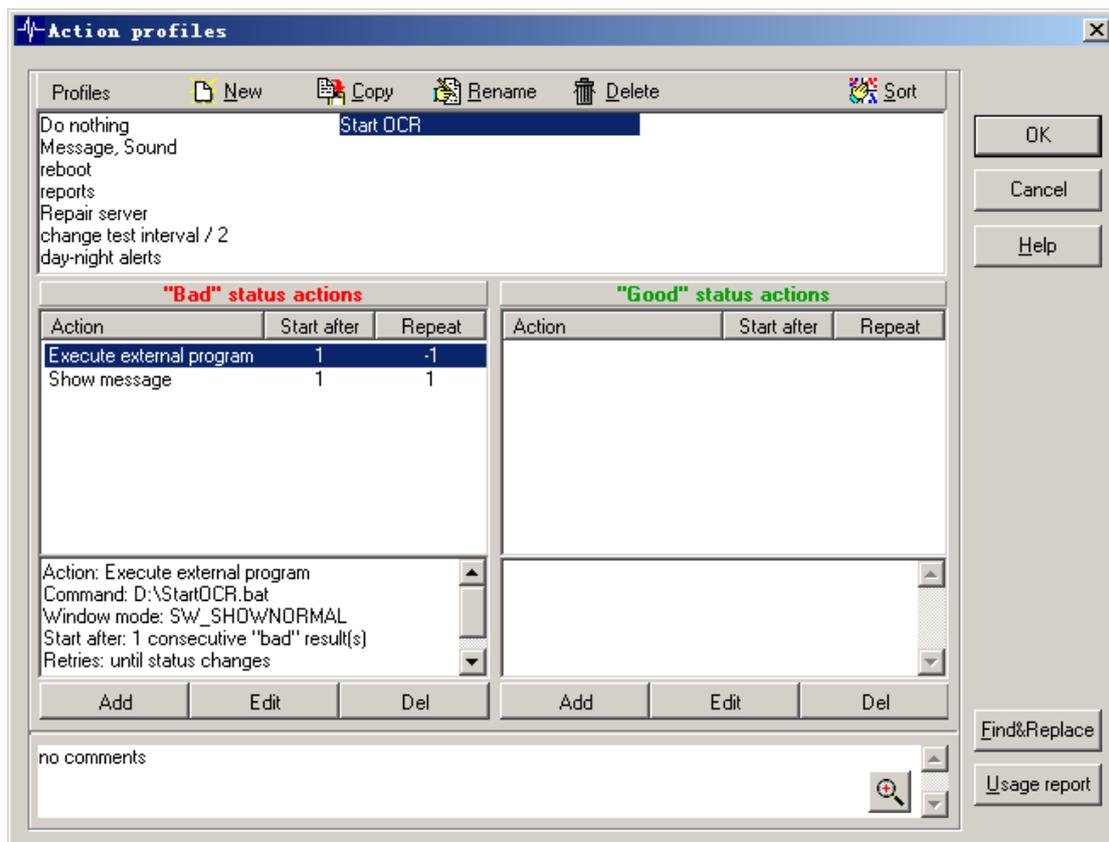
```
start cmd /k "D:\yunmai\IDcard.exe"
```

其中 D:\yunmai\IDcard.exe 即为程序的绝对路径。

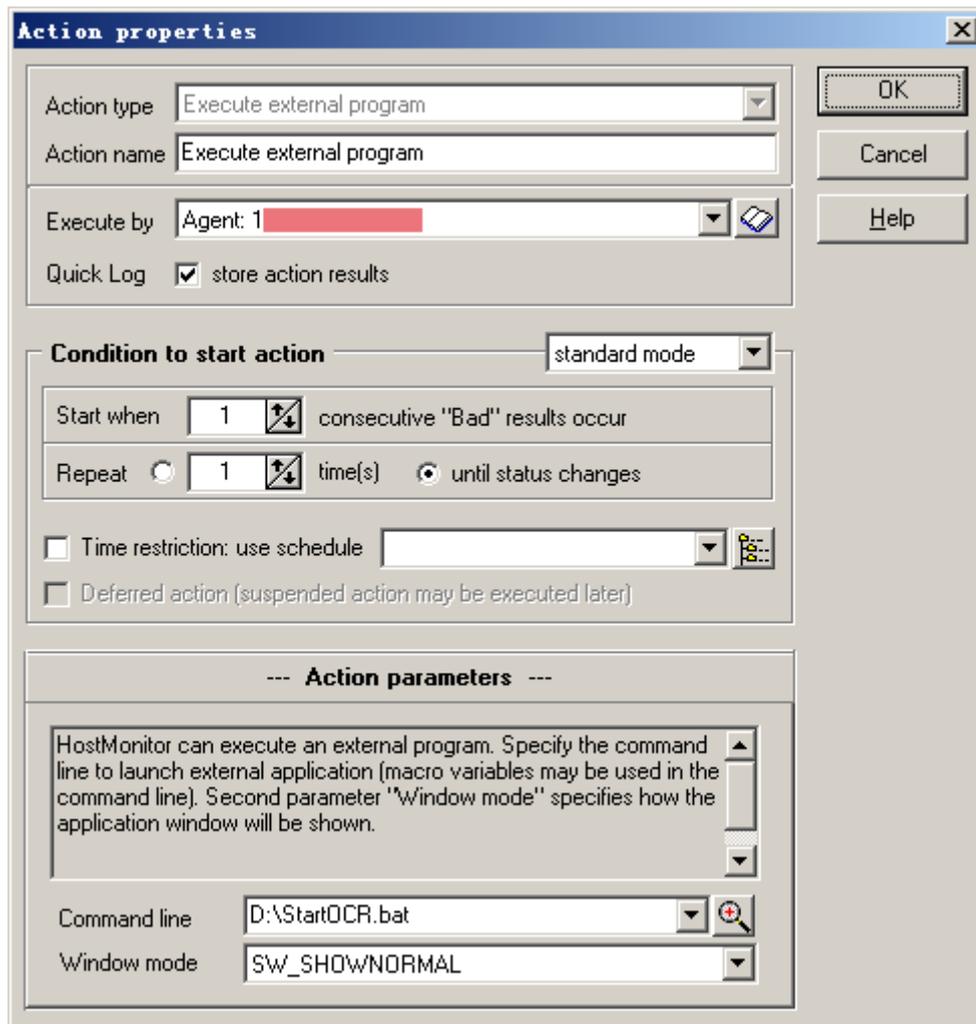
第二步，按照上一节的方法添加对进程的监控：



第三步，创建个性化告警 Start OCR，并添加执行外部程序（Execute external program）和显示消息弹窗（Show message）两个动作：



Execute external program 按照如下图设置，注意 Execute by 需要设置为目标服务器上的 RMA，Repeat 这里设置为直到状态改变，最重要的是 Command line，设置为我们第一步中创建的 Bat 批处理文件：

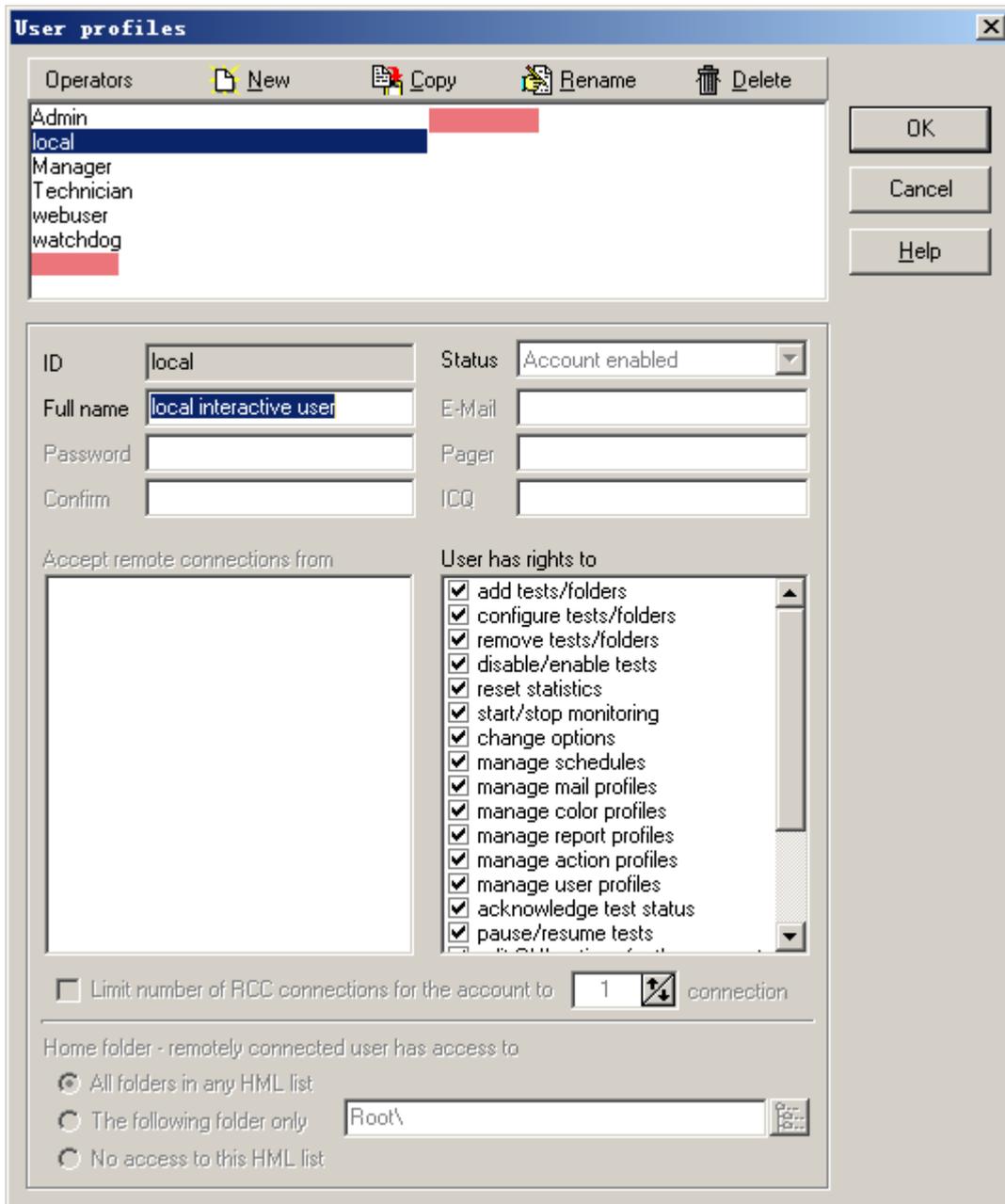


如此便大功告成，可以手动测试下效果了。Hostmonitor 会在 IDcard.exe 进程退出后自动弹出消息告警并重启程序，完全不需要人工干预，而且可以二十四小时运行，不必操心进程挂掉的感觉非常爽，我们又可以解放出来喝喝茶……哦不对，从事更需要创造力的工作了:-)

## 10 RCC 远程管理 HostMonitor

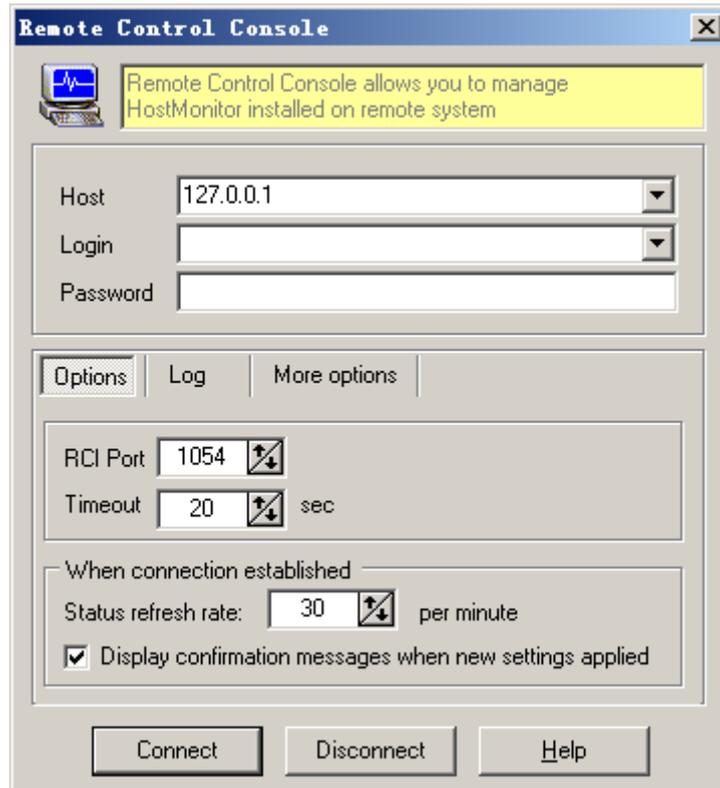
RCC 全称是 Remote Control Console，即 HostMonitor 的远程管理控制台，它可以连接到远程主机上的 HostMonitor，就像本地运行的 HostMonitor 一样管理和报警。有什么用呢？一般系统管理员不止一名，没必要每个人都运行一套 HostMonitor，这样维护起来并不方便，重复监控也会提高对系统的性能压力，这时候 RCC 就能派上很好的用场。另外跨网段监控的时候，使用 RCC 也会很合适。

要使用 RCC 必须先配置 HostMonitor 中的用户，打开 User 中的 Operators 菜单，就进入了操作员管理界面：



可以创建、修改操作员，并赋予不同的权限，还可以限制登录 IP。操作非常简单，不再细说。

创建操作员后，我们就可以从远程主机上登录 RCC 了！RCC 的登录界面是这样的：



使用也非常简单，用设定的用户名和密码登录系统后，就能像本机一样管理 HostMonitor 的各项功能，也同样能实现声光报警，特别适合多人协同工作。

## 11 后记

HostMonitor 这样的工具让我们能在第一时间发现问题，这对系统运维工作有着重要的意义。HostMonitor 的功能十分丰富，本文只是我在国元证券从事系统管理工作的多年实践中探索积累出来的一些经验的记录，相信还有更多更有用的玩法有待发现和探索。时至今日，HostMonitor 已经成为我和我的团队在系统管理工作中离不开的重要助手。我的个人电子邮箱是 [nuoyan\\_cfan@163.com](mailto:nuoyan_cfan@163.com)，欢迎各位朋友来信交流。最后送上让我深有感触的两句话，给从事系统运维管理工作的朋友们共勉：

良医者，常治无病之病，故无病。圣人者，常治无患之患，故无患。

——《淮南子·说山训》

管理得好的工厂，总是单调乏味，没有任何激动人心的事件发生。

——彼得·德鲁克

## 12 附录：HostMonitor 测试方法介绍

本节内容翻译自 HostMonitor 帮助文档。

### 12.1 Ping

Ping 命令通过 ICMP 协议来检测远程主机、路由器或其他网络设备的连通性，是最佳的通用网络连通性检查方法。

### 12.2 Trace

追踪到达远程主机的路由。这个方法允许你检查从本地主机到指定的互联网（或局域网）主机之间的数据包路由。也能用于检查每一跳花费的响应时间。

### 12.3 URL

HostMonitor 能通过 HTTP、HTTPS、FTP 或 Gopher 协议发起 URL 请求。HostMonitor 可以直接向远程主机发起请求，也可以通过代理服务器发起。

### 12.4 HTTP

HTTP 检测方法提供了监控 HTTP 服务器的能力，能够校验动态页面的内容，或是检查内容是否发生变化。相比同样可以执行 HTTP 请求的 URL 检测方法，HTTP 检测方法不会使用 wininet.dll（IE 浏览器的组成部分），这样能够避免一些 IE 自身的 BUG（当然我们也可能犯错，但我们总有能力修正自己犯的错误）。另外使用 HTTP 检测方法你也能够定义更多的参数进行更多样的检查。

### 12.5 SOAP/XML

SOAP 是 Web 服务的各种实现中专门用于交换结构化信息的一种协议。它使用 XML 作为消息格式，通常使用 HTTP 或是 SMTP 协议实现消息传输。

这个测试方法允许通过 HTTP 或 HTTPS 协议执行 SOAP 协议的 1.1 或 1.2 版本。

HostMonitor 支持 GET 或 POST 请求，支持客户端或服务器端证书，还可以自定义 HTTP 头。

## 12.6 Certificate expiration 数字证书有效期

这个测试方法允许你检查 SSL 数字证书的到期日，并在指定的天数之前报警。它设计用于 HTTPS 服务器，但也可以用于隐式支持 SSL 的服务器，比如一些 SMTP、POP3、IMAP 服务器（但不是全部）。

## 12.7 Domain expiration 域名有效期

检查域名的到期日，并在指定的天数之前报警。

## 12.8 SMTP

检查 SMTP 服务器状态。互联网上的绝大部分 E-Mail 服务器都是用 SMTP 协议在服务器之间传输电子邮件。HostMonitor 假装为电子邮件客户端或另一台校验用户存在性的服务器来发送 SMTP 命令到目标服务器来实现监控。通过校验用户名，程序有效地检测 SMTP 服务器的邮件数据库而不必产生非必要的邮件消息。

## 12.9 POP3

检查 POP3 服务器状态。通常电子邮件使用 SMTP 在邮件服务器之间传送，但读取电子邮件需要使用 POP3（Post Office Protocol version 3）协议。POP3 检测方法能够用来测试你的邮件服务器的 POP 功能是否正常工作。HostMonitor 使用指定的用户账户登录并登出服务器。另外 HostMonitor 也可以检查邮件数量或指定用户的邮件大小，如果数量或大小超过了限制数，程序能自动报警。

## 12.10 IMAP

检查 IMAP 服务器状态。一些邮件服务器支持 IMAP 协议（Internet Message Access Protocol）用于交换和处理邮件。IMAP 检测方法能够像 POP3 检测方法一

样检测 IMAP 邮件服务器。

## 12.11 E-Mail 电子邮件内容

检查电子邮件内容。使用 POP3 或 IMAP 协议检测目标服务器上的电子邮箱，支持加密连接（支持的 SSL 协议包括 TLS1、SSL3、SSL2 和 PCT1）。可以用来校验收件人地址、发件人地址、邮件标题或邮件正文。

## 12.12 MailRelay

监控邮件服务器链。HostMonitor 通过特定邮件服务器发送一封电子邮件，并在邮件到达目的邮箱时进行检查。HostMonitor 使用 SMTP 协议发送邮件，并使用 POP3 或 IMAP 协议检查目的邮箱。

## 12.13 TCP

监控基于 TCP 协议的服务器，例如 Whois、Finger、Telnet、Charge 等等。注意如果你需要检查 SMTP、POP、IMAP、DNS、LDAP、HTTP、HTTPS、FTP 服务器的话建议使用特定的检测方法。

## 12.14 UDP

监控基于 UDP 协议的服务器。例如 TFTP、SNTP、Daytime 等。注意，如果你需要检查 DNS、NTP、RADIUS 服务器的话建议使用特定的检测方法。

## 12.15 NTP

监控 NTP 服务器状态。NTP 协议在全球互联网中广泛用于同步计算机时间，使用 NTP/SNTP 检测方法能够很容易地检查 NTP/SNTP 服务器状态是否正常。

## 12.16 DNS

监控 DNS 服务器状态。DNS 检测方法直接连接到指定的 DNS 服务器并发起

DNS 查询，也就是说完全不受 host 文件、DNR 缓存之类的影响。

## 12.17 DHCP

监控 DHCP 服务器状态。HostMonitor 通过发送请求来校验 DHCP 服务器状态，只要 DHCP 服务器在超时时间内返回了消息请求，无论是 NAK 或 ACK 包，HostMonitor 都会将 DHCP 服务器标记为存活状态。否则就会标记为 Bad（如果返回值无效）或 No answer 状态。

## 12.18 LDAP 目录服务器

使用 LDAP 协议检测目录服务器状态。执行这个测试方法 HostMonitor 需要连接并绑定到目录服务器。

## 12.19 RADIUS

用于检测 RADIUS 服务。

## 12.20 DICOM

监控 DICOM 服务器状态。医用数字图像和通讯标准 Digital Imaging and Communications in Medicine (DICOM) standard 由 NEMA 创建，用于辅助医学图像的分发和展示，例如 CT 扫描、MRIs、超声等。

## 12.21 RAS

监控远程 RAS 服务器状态。

## 12.22 UNC 共享资源可用性

监控 UNC（Universal Naming Convention，通用命名规范，用于网络共享）或本地文件夹路径的可用性，检查可用空间总数，或是检查当前用户的可用空间数（如果操作系统启用了用户的磁盘配额功能）。检测方法可以选择以下条件进行

报警:

- 1、 资源不可用
- 2、 当前用户可用空间小于指定值
- 3、 总可用空间小于指定值

如果需要通过安装在类 UNIX 系统中的 agent 中使用, 则需要注意路径中应当使用斜杠 (/) 而不是 Windows 系统中使用的反斜杠 (\), 并且文件名要注意大小写。

## 12.23 Drive Free Space 磁盘可用空间

监控可用磁盘空间容量。这个检测方法和 UNC 检测方法都能用于检查本地或远程系统的可用磁盘空间, 但他们是基于不同的方式和协议实现的。

单个 UNC 检测方法使用文件系统请求可以检查单个本地或共享网络资源。

单个 Drive Free Space 检测方法可以检测目标系统中的一系列驱动器并找出可用空间最小的驱动器, 如果需要甚至可以加上可移动磁盘。检查远程系统时由于使用了 WMI 请求, 所以可以检查远程 Windows 主机上的非共享资源。

## 12.24 Folder/File Size 文件夹或文件大小

监控文件或文件夹大小。输入文件夹地址和最大大小即可使用。可以手动输入完整地址, 也可以点击浏览按钮选择文件。

如果选中了“包含子文件夹 (Include sub-folders)”参数, HostMonitor 会统计所有子文件夹的大小。

如果需要检查动态创建、没有固定文件名的文件或文件夹 (比如每天创建的日志文件), 你可以开启“转换宏命令 (Translate macros)”选项, 这样就可以在路径或文件名中使用特有的日期宏变量、文件变量以及用户定义的变量等。

如果需要通过安装在类 UNIX 系统中的 agent 中使用, 则需要注意:

- 1、 路径中使用斜杠 (/) 而不是反斜杠 (\)。
- 2、 类 UNIX 系统对文件名的大小写敏感。
- 3、 在 Windows 系统中 “\*” 和 “\*.\*” 表示任意文件, 类 UNIX 系统中只有 “\*” 表示任意文件, 而 “\*.\*” 表示任意文件名中含有 “.” 字符的文

件，二者并不相同。

## 12.25 Count Files 计算文件数量

计算指定目录下的文件数量。

## 12.26 Folder/File Availability 文件夹或文件可用性

监控文件夹或文件的可用性。可以设置文件必须**存在**，也可以设置文件必须**不存在**，不符合条件时执行报警动作。如果需要监控动态生成的文件名，则可以使用宏变量。

可以设置扩展参数，只在文件太旧（或太新）超过指定时长后报警。

如果需要通过安装在类 UNIX 系统中的 agent 中使用，则需要注意：

- 1、 路径中使用斜杠 (/) 而不是反斜杠 (\)。
- 2、 类 UNIX 系统对文件名的大小写敏感。
- 3、 在 Windows 系统中 “\*” 和 “\*. ” 表示任意文件，类 UNIX 系统中只有 “\*” 表示任意文件，而 “\*. ” 表示任意文件名中含有 “.” 字符的文件，二者并不相同。

## 12.27 File Integrity 文件完整性

监控文件的完整性。HostMonitor 能够在文件发生变化时报警。原理是计算文件的 CRC 校验值，并在每次监控时检查 CRC 校验值是否保持一致。点击 “计算 CRC (Calculate CRC)” 按钮程序将为文件计算并保存 CRC 校验值。

## 12.28 Text Log 文本日志内容

监控文本日志文件。在监控由其他程序创建的日志文件时非常有用。不同于文件的任意部分出现指定字符都会报警的 “文件比较 (Compare Files)” 方法，Text Log 方法只会在新增记录行中出现指定字符时才会报警。这意味着如果你在启动 HostMonitor 时日志文件中已经有了符合报警条件的记录行时，你不会再收到任何报警；但是如果在 HostMonitor 运行时有任何新的符合报警条件的记录行

被加入到日志中，你都会再次收到告警信息。也就是说 Text Log 的工作方式很像 NT 事件日志（NT Event Log）方法，只不过它监控的是文本文件而不是 NT 系统的事件日志数据库。

## 12.29 Compare Files 文件比较

比较两个文件，或是在文件中查找特定字符串。可以设置 6 种告警条件：

- 1、 两个文件不同时报警
- 2、 两个文件完全相同时报警
- 3、 第一个文件的任意部分中包含第二个文件时报警
- 4、 第一个文件的任意部分中不包含第二个文件时报警
- 5、 文件包含特定字符时报警
- 6、 文件不包含特定字符时报警

可以指定一或多个比较方法：比较时间、比较文件大小、比较文件内容。如果选择了比较文件内容但没有选择比较文件大小，HostMonitor 会认为其中之一从 0 字节开始就包含了另一个文件所有内容的两个文件是完全相同的（也就是说一个文件可以比另一个文件小）。

可以使用宏变量。

如果用来检查文件中是否包含特定字符串，则需要告诉 HostMonitor 这个文件使用的字符编码，HostMonitor 支持的编码方式包括 ASCII、UTF-8、UTF-16、UTF-16 big endian、UTF-32、UTF-32 big endian。

## 12.30 Process 进程

监控进程数量。简单意义上来说，一个进程就是一个可执行程序。HostMonitor 能够控制本地或远程主机上的特定进程，并在实例或特定进程的数量超出设定的范围后报警。控制远程主机上的进程需要有相应的管理员权限。

## 12.31 Service 服务

监控服务状态。微软的 Windows NT 系统支持一种叫做“服务”的应用程序

类型。服务可以在系统开机时自动启动，也可以由用户在服务控制台中手动开启，还可以由应用程序调用服务控制函数启动。服务甚至可以在没有用户登陆到系统的情况下执行。

这里 Windows NT 表示基于 NT 技术的 Windows 操作系统，包括 Microsoft Windows NT 4.0、Windows 2000、Windows XP、Windows Vista、Windows Server 2003、Windows Server 2008、Windows 7 以及之后的 Windows 系统。

为了监控服务状态，HostMonitor 会建立到指定服务器的服务控制管理器的连接，并请求服务立即更新自身状态。如果服务尚未启动，或者没有应答，则 HostMonitor 会执行指定的报警动作。注意：监控远程计算机上的服务状态需要有相应的权限。

## 12.32 NT Events Log 事件日志

监控 Windows 事件日志。微软 Windows 系统的事件日志功能提供了标准的、集中式的应用程序（和操作系统）记录重要软硬件事件的方法。事件日志服务将不同来源的事件存储在一个单一的数据集中，叫做事件日志。HostMonitor 能够监控日志文件并在特定的事件发生时报警。

## 12.33 CPU Usage CPU 利用率

监控本地或远程主机上的 CPU 使用百分比。支持下列操作系统：Microsoft Windows NT 4.0、Windows 2000、XP、Windows Server 2003、Vista、Windows Server 2008、Windows 7 and 8、Novell Netware 4+ 以及使用 RMA 程序的 AIX、BSD、Linux 和 Solaris 操作系统。

## 12.34 Memory 可用内存

监控本地或远程主机上可用的物理内存或虚拟内存大小，或是未使用的交换空间（页面文件）容量。支持 Windows、Linux、BSD 操作系统、思科路由器以及一些其他网络设备。

## 12.35 Performance Counter 性能计数器

监控本地或远程计算机上通过系统或服务执行的任务的性能特征。可以用于监控 Windows NT 系统中许多的重要参数。

## 12.36 WMI

通过 WMI 监控 Windows 系统。WMI 是 Windows Management Instrumentation 的缩写，是微软对基于 WEB 的企业管理 Web-Based Enterprise Management (WBEM)——一种新的管理技术，允许软件监控和控制遍布在网络中的各种可管理资源——的一种实现。可管理的资源包括硬盘、文件系统、操作系统设置、进程、服务、共享、注册表设置、网络组件、事件日志、用户、用户组等等。

WMI 同样可以监控性能计数器。包括 Exchange 和 SQL Server 在内的微软应用程序都内建了 WMI 能。一些其他厂商出品的软件也支持 WMI，它们同样可以使用 HostMonitor 来监控。

WMI 在 Windows 2000 和更高版本中提供。

## 12.37 Registry 注册表

监控本地或远程主机上的注册表键值。

## 12.38 Dominant Process 主要进程

找出本地或远程主机上资源使用率最高的进程。

和其他允许你监控指定进程或是资源使用率的方法不同，主要进程检测方法允许你找出哪个进程使用了最多的系统资源，比如 CPU、句柄、线程、内存、虚拟内存等等。HostMonitor 能提供顶级进程的信息，包括分配的资源数量、进程名称或进程 ID 等。

## 12.39 VM host status 虚拟主机状态

监控 VMware ESXi 或 Microsoft Hyper-V 主机的健康状态。

## 12.40 VM host CPU usage 虚拟主机 CPU 利用率

监控虚拟主机的 CPU 利用率。

## 12.41 VM host free memory 虚拟主机内存

监控虚拟主机的可用内存百分比。

## 12.42 VM host free datastore 虚拟主机可用空间

监控虚拟主机数据存储空间，并找出可用空间最小的存储器。

## 12.43 VM guest status 虚拟机状态

检查运行在 VMware 或 Microsoft Hyper-V 系统上的所有虚拟系统或客户机，当客户机的心跳或健康状态显示为失败时报警并显示客户机名称。

## 12.44 VM guest CPU usage 虚拟机 CPU 利用率

检查运行在 VMware 或 Microsoft Hyper-V 系统上的所有虚拟系统或客户机，找到 CPU 使用率最高的客户机。

## 12.45 VM guest free memory 虚拟机可用内存

检查运行在 VMware 或 Microsoft Hyper-V 系统上的所有虚拟系统或客户机，找到可用内存最小的客户机。

## 12.46 VM guest free disk space 虚拟机可用磁盘空间

检查运行在 VMware 或 Microsoft Hyper-V 系统上的所有虚拟系统或客户机，找到可用硬盘最小的客户机。

## 12.47 Database Server 数据库服务器

监控数据库服务器状态。采用登录、登出方式检测数据库，支持 Interbase、SQL Server、MySQL、Oracle、Postgre、Sybase Server。需要提供服务器名称（Oracle 需提供 TNS 别名）、数据库名、用户名和密码。Interbase 需要指定协议（TCP、SPX 或 NetBEUI），检测 MySQL 或 PostgreSQL 时可以修改默认端口号。

监控 Oracle 数据库时 HostMonitor 需要使用 Oracle Call Interface（OCI.DLL），所以应当安装 Oracle 客户端软件。其他数据库也需要必要的客户端软件。HostMonitor 用到下列动态链接库：

- gds32.dll：用于监控 Interbase
- ntwdblib.dll：用于监控 SQL Server
- libmysql.dll：用于监控 MySQL
- libpq.dll：用于监控 PostgreSQL
- libsybdb.dll：用于监控 Sybase

如果客户端组件没有安装导致无法执行检查时，可能会显示为“未知（unknown）”状态。

## 12.48 ODBC Query 执行 SQL 查询

监控 ODBC 数据源的可用性，还可以执行 SQL 查询，并分析返回数据集中的指定字段的值。设置这个测试方法时，只需选中系统中已经预先定义好的 ODBC 数据源，然后填写登录用户名、密码和超时时间字段即可。还可以选择执行 SQL 查询并设置监控条件，如果没有指定任何监控条件，则可能显示两种状态：Alive 或 No answer。如果指定了监控条件，监控方法的状态可以为下列的其中一种：

- OK：如果条件匹配
  - Bad：如果条件不匹配
  - Unknown：如果查询失败，或在返回结果中找不到指定的字段
- 你也可以手动设置返回结果中找不到指定字段时显示的状态。

### Windows x64 的特殊说明

HostMonitor 是 32 位应用程序，但可以在 64 位系统中正常运行。如果你使

用的是 64 位 Windows 系统,你应该使用 32 位的 ODBC 数据源管理器来设置 ODBC 数据源 (一般在 C:\WINDOWS\SysWOW64\odbcad32.exe)。

## 12.49 SNMP Get 获取 SNMP 信息

读取 SNMP 信息。SNMP (Simple Network Management Protocol, 简单网络管理协议) 是一种用于在管理控制台程序和可管理实体 (主机、路由器、网桥、交换机等设备) 之间交换管理信息的互联网标准协议。使用这个测试方法 HostMonitor 能够控制各类网络设备的各种参数。

## 12.50 SNMP Trap SNMP 陷阱

接收 SNMP 陷阱消息,即网络设备主动发送到 SNMP 控制台的消息。陷阱 (Traps) 能够指示电源或线路连接状态、温度超出设定值、过高的网络流量等问题。Traps 提供了一种即时事件通知机制。

这个检测方法与其他方法不同。不像一些通过发送请求和接收响应的方式真正检测设备的方法,SNMP Trap 检测方法作为监听器而实现——它并不发送任何请求。HostMonitor 监听来自网络主机的消息,并即时响应。

因此,时间间隔字段在这个检测方法上不可用。但是,你仍然可以像其他测试方法一样使用任务计划。HostMonitor 会在非工作时段内自动忽略来自网络设备的消息。

此外,Active RMA agent (主动模式的 RMA) 能够收集、过滤、转发 SNMP 陷阱消息到 HostMonitor,有些地方需要注意:

- 1、RMA 和 HostMonitor 之间的网络流量会被加密,转发的陷阱消息也是。
- 2、当在 Test by 中选择远程的主动模式 RMA 时,HostMonitor 会发送过滤设置到 RMA。这允许过滤远程站点的消息,只将符合指定条件的消息发送给 HostMonitor。
- 3、SNMP Trap 检测方法中无法使用“备用 RMA (Backup RMA)”

## 12.51 SNMP Table 批量获取 SNMP 信息

每个 SNMP Get 测试项一次只能检查 1 个计数器，但每个 SNMP Table 测试项一次就可以检测数以百计的计数器。单个 SNMP Table 无法同时检测多个 SNMP 代理，但是可以读取目标中的计数器集合并执行不同的检测。例如你可以设置 HostMonitor 在同一个 Table 中找到最大或最小的计数器，检查平均值或找出变化最剧烈的计数器。有些时候用一个 SNMP Table 就足以替换数以千计的 SNMP Get 检测项。例如如果你想在路由器的任何网络端口发生状态变化时都能收到告警，就可以简单地使用一个 SNMP Table 来实现。

## 12.52 Traffic Monitor 网络流量监控

监控启用了 SNMP 的设备上的网络接口流量。相比 SNMP 检测方法，它有些优势：

- 1、 不需要知道 OIDs，设置界面会自动显示可用的接口，你只需要选择你想监控的网络接口。
- 2、 单个测试项即可监控接口上的整体网络流量，包括流入量和流出量。
- 3、 通过单个测试项就能监控指定设备所有接口的网络流量或其他参数例如队列长度的整体情况。例如你可以监控一台路由器上所有端口的流量。

## 12.53 Temperature Monitor 温度监控器

如果使用了 Sensatronics 的温度传感器单元，这个监控方法允许在温度超出指定范围时报警。

## 12.54 Active Script 活动脚本

从 HostMonitor 3.4 版本开始可以使用不同的脚本语言比如 Visual Basic script 或 Java script 创建属于自己的测试方法。HostMonitor 使用微软的 ActiveScripting 技术执行由你自己或其他人编写的脚本程序。理论上使用这个测试方法可以监控

任何需要监控的东西（如果有些信息不能通过脚本语言取回，你也能用 C++ 或者其他语言写个 ActiveX 对象并在脚本里调用）。这样，你就能够改进 HostMonitor 来满足你的所有需求。另外说一句：你甚至还能创建属于自己的脚本程序语言。

## 12.55 Shell Script Shell 脚本

Shell 脚本检测方法结合了“外部程序 (External)”方法的广泛用途和“活动脚本 (Active Script)”方法的易用性。“外部程序 (External)”和“Shell 脚本 (Shell Script)”都将指定的命令作为新进程执行。由于这种特性，使用任何编程工具创造检测方法都成为可能。例如，你可以使用 Visual Basic script、Java script、使用 C++ 编译器创造真正的可执行文件或者只是写个简单的批处理或 shell 脚本。反过来说这个检测方法也允许任何无效的状态或响应值返回到 HostMonitor，正如“活动脚本 (Active Script)”检测方法一样。

## 12.56 External 外部测试

HostMonitor 内置了大量的检测方法，但不可能支持所有的检测类型。这就是为什么 HostMonitor 包含了“外部测试 (External Test)”。使用外部测试方法，HostMonitor 能够调用其他应用程序执行测试，并依据返回的错误级别将测试状态标记为成功或失败。

## 12.57 SSH

SSH 是一种通过安全渠道在两台网络设备之间传输数据的网络协议，主要用于类 UNIX 系统访问 Shell 账户，设计用来替代 TELNET 和其他不安全的远程 shell。SSH 典型的应用是用来登录远程计算机并执行命令。它的加密机制能够保证数据在不安全的网络（比如 Internet）上传输时的可信性和完整性。

HostMonitor 能够连接、登入指定的远程系统并执行 1 条命令或 shell 脚本。因此允许你在不使用 RMA 的情况下在 UNIX 系统中执行一些测试方法。

HostMonitor 可以直接执行 SSH 检测方法，或是通过 Windows 上的 RMA 来执行。

## 12.58 HM Monitor

试想一下，你在服务器上安装了一套 HostMonitor 并且设置了数以千计的测试项来监控整个网络，HostMonitor 会显示出系统中的任何问题，除非……除非问题刚好发生在 HostMonitor 运行的主机上。供电中断、主板烧坏或者系统崩溃了，怎么办？你可以安装在另一个系统中再安装一套 HostMonitor 程序，并且使用备用 HostMonitor 来监控你的主用 HostMonitor！这样做还有一个好处：主用 HostMonitor 也会监控备用 HostMonitor 所以每套系统都能同时监控！

注意：

- 1、 如果你想在两套系统中分别安装 HostMonitor（不管是物理机还是虚拟机），你都需要购买 2 套授权。
- 2、 为了使用“B”来监控“A”，你需要在“A”上启用 RCI。如果你希望使用两台 HostMonitor 分别监控对方，则需要在两台 HostMonitor 上同时启用 RCI。RCI 授权只在企业版中免费提供，简化版、标准版或专业版用户需要单独购买。
- 3、 企业版中同样提供了 WatchDog 程序，允许你远程监控 HostMonitor 状态并及时报警。WatchDog 提供了图形化和警报功能并且无需额外成本，但是两套 HostMonitor 实例则能够提供最大的灵活性和可靠性。

HM Monitor 测试方法不仅仅提供简单的在线状态检测功能，还能监控 HostMonitor 的大量参数，比如：

能够在监控项目被禁用或是停止监控时报警；

能够显示主监控程序每秒钟执行多少个测试项，并在指定周期内执行的测试项低于 5 个时报警；

能够指出你的测试列表被修改但是还没有被保存；

能够在 ODBC 驱动花费了太多时间时报警；

能够告诉你主监控程序执行了多少个动作，以及记录了多少条日志；

以及更多。