CA Spectrum®

分布式 SpectroSERVER 管理员指南

版本 9.3

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 3 章： 在分布式 SpectroSERVER 环境中跨防火墙进行通信 53

第 4 章： 容错 61

第 5 章： 处理陷阱定向器 77

第 1 章：分布式 SpectroSERVER 简介

此部分包含以下主题：

关于分布式 SpectroSERVER

分布式 SpectroSERVER (DSS) 是一种强大的建模功能，它允许在大规模基础架构中分配管理任务。基础架构可以按地理位置或者跨单个物理位置中的多个服务器进行组织。在管理计算基础架构时，DSS 可以提高 CA Spectrum 性能。性能提高可以归因于分配管理通信产生的网络负载以及将任务委派给远程工作站。

使用 DSS，可以创建由多个格局组成且每个格局都具有本地 SpectroSERVER 的基础架构的统一表示。在 DSS 环境中，SpectroSERVER 客户端（如 OneClick 控制台）可以同时访问多个 SpectroSERVER 的信息。

DSS 包括以下功能：

分布式可见性

OneClick 控制台显示所有已部署 SpectroSERVER 的信息。为每个 SpectroSERVER 显示摘要警报计数。网络管理员可以快速找到故障区域。

本地化的轮询通信

正在轮询网络的管理工作站在地理位置上更接近它通过 DSS 管理的设备。此设置可减少广域链路上的通信并避免本地网络阻塞。与轮询远距离设备的单个 SpectroSERVER 相比，多个更小的 SpectroSERVER 生成的通信量更少。

可伸缩性

随着网络的扩展，将低端工作站用作其他专用 SpectroSERVER 服务器的代价通常较小。在大型非-分布式环境中升级一个工作站以适应一个 SpectroSERVER 时，这样的解决方案更为可取。

容错

DSS 支持 SpectroSERVER 之间的容错。可以提供一个备用 SpectroSERVER 来作为主要 SpectroSERVER 的容错备份或待机元素。运行主要 SpectroSERVER 的工作站失败时，网络管理能够继续。

只需将给定格局的数据库重新加载到备用 SpectroSERVER 上，即可防止发生故障。当发生故障导致主要 SpectroSERVER 被禁用时，备用 SpectroSERVER 将会自动替代它。所有的 CA Spectrum 应用程序都将自动使用备用 SpectroSERVER。

详细信息：

建立容错 (p. 68)

格局

单个 SpectroSERVER 所管理的网络域称为*格局*。格局由模型、关联、属性值、警报、事件和属于特定 SpectroSERVER 的统计信息组成。网络中的每个格局都是唯一的。唯一的格局句柄 (ID) 标识每个格局。

在 OneClick 中，每个格局由一个格局图标表示。格局图标表示 SpectroSERVER 数据库。通过双击区域和菜单选择，格局图标允许您访问远程网络模型。该图标还提供在那些远程数据库中建模的设备的警报信息汇总。

注意：术语*本地*和*远程*用于表示从特定 SpectroSERVER 的角度提及格局。

格局映射

CA Spectrum 自动维护构成特定 DSS 环境的所有格局的“映射”。在 SpectroSERVER 拓扑中发生的更改会启动内部发现机制，从而更新该环境中所有其他 SpectroSERVER 的格局映射。例如，发现能检测网络中 SpectroSERVER 的添加或删除情况。

格局模型是一个容器模型。格局模型允许您通过 CA Spectrum 用户界面连接到格局映射中的其他 SpectroSERVER。可以在以下三个视图层次结构的任何级别上创建格局模型：拓扑、位置或组织。

建模目录

建模目录是在特定的 SpectroSERVER 上安装的一组模板(如模型类型或关系)。这些模板用于创建模型。通过这些模板创建的模型构成该 SpectroSERVER 的格局。

注意：OneClick 在启动时会查询默认格局以获取可用模型类型的列表。

在使用 DSS 对多个格局建模时，每个格局都必须包含相同的建模目录。一个格局的建模目录中的所有模型类型也必须存在于它所连接到的每个其他格局的建模目录中。如果在一个 SpectroSERVER 上安装了新的管理模块，请在格局映射中每个其他 SpectroSERVER 上安装相同的管理模块。

用户模型

用户模型包含有关单个用户权限和其他用户数据的信息。此信息存储在格局中。首次安装 CA Spectrum 时，默认用户模型表示 CA Spectrum 安装期间在“安装所有者”字段中指定的用户。

注意：有关详细信息，请参阅*《管理员指南》*。

在 DSS 环境中，相同的用户模型必须存在于所有格局中，才能使用户可以连接到格局映射中的远程 SpectroSERVER，以及使格局可以相互通信。添加或修改用户模型会导致与格局关联的 SpectroSERVER 查询格局映射中的所有其他 SpectroSERVER。该查询将检查其他格局中的用户模型。如果在其他格局中找到用户模型，则将自动更新其用户模型，以反映对初始用户模型的任何更改。

注意：删除某个用户模型仅会在该格局中删除它。必须手动删除格局映射中其他 SpectroSERVER 上的重复用户模型。

SpectroSERVER (.vnmrc) 资源

SpectroSERVER 资源在 <*$SPECROOT*>/SPECTRUM/SS/.vnmrc 文件中定义。这些资源文件的许多默认值内置到代码中并留空。但是，在这些资源中指定值时，不使用编码的默认值。

.vnmrc（虚拟网络计算机运行时配置）文件中的资源定义 SpectroSERVER 的路径名和默认设置。CA Spectrum 系统软件包括具有 SpectroSERVER 资源的默认设置的运行时配置文件。

可以修改资源以进行以下更改：

- 定义常规 SpectroSERVER 资源和目录路径
- 调整事件存档
- 指定命名服务变量
- 调整线程分配
- 控制容错警报同步

所有这些资源条目都以“resource = resource_value”格式列出。对于 .vnmrc 文件中的许多资源，资源值为空。

如果未显示特定资源的值，则 CA Spectrum 使用默认值。.vnmrc 文件中的资源在启动 SpectroSERVER 后生效。如果在 SpectroSERVER 正运行时更改此文件，则所做的更改在重新启动 SpectroSERVER 后生效。

示例：定义 .vnmrc 资源

此示例设置在事件日志数据库中记录的最大记录数：

```
max_event_records=5000
```

max_event_records

指定资源名称。

5000

指定资源值。

常规 SpectroSERVER (.vnmrc) 资源

常规 SpectroSERVER (.vnmrc) 资源控制许多 SpectroSERVER 功能。以下列表介绍了常规 SpectroSERVER (.vnmrc) 资源：

comm_port

指定客户端用户界面用来与 SpectroSERVER 通信的 TCP 端口。端口号在安装期间定义。

该命令采用以下格式：

comm_port=*0xBEEF*

0xBEEF

指定默认的 TCP 连接端口套接字。此套接字可以是大于在文件 /netinet/in.h 中分配给 IPPORT_USERRESERVED 参数的端口号的任何有效 TCP 端口。但是，端口必须小于 65535 (0xFFFF)。

expiration_date

指示 CA Spectrum 许可证的到期日期。在安装期间输入的 CA Spectrum 密钥包含到期日期。

该命令采用以下格式：

`expiration_date=`*mm/dd/yyyy*

密码

用安装期间输入的 CA Spectrum 密钥自动填充。在安装之后，请勿更改此资源。

该命令采用以下格式：

`password=`*password*

密码由 CA 提供。

snmp_comm_port

指定可通过 SpectroSERVER 从其发送 SNMP 请求的端口。

该命令采用以下格式：

`snmp_comm_port=`*port#*

port#

设置为介于 0x400 (1,024) 和 0xFFFF (65,535) 之间的任何无符号 16 位整数。SNMP 代理的一些实施（如 IBM Mainframe MVS 系统）将端口视为有符号数。在这些情况下，请将此资源设置为介于 0x400 (1,024) 和 0x7FFF (32,768) 之间的值。

默认值： 0xFFFF

snmp_trap_port_enabled

将 SpectroSERVER 绑定到 SNMP 陷阱端口并侦听陷阱。设置为 False 时，SpectroSERVER 不绑定到 SNMP 陷阱端口。

该命令采用以下格式：

`snmp_trap_port_enabled=`*TRUE*

默认值： True

vnm_file_path

指定包含 SpectroSERVER 外部文件（如数据库文件）的根子目录。

该命令采用以下格式：

`vnm_file_path=`<*目录*>*/spectrum/SS/CsVendor*

/spectrum/SS/CsVendor

指定根子目录的文件路径

tcp_buffer_size

可用于更改适合于 LAN 上通信的缓冲区大小。大的缓冲区大小可提高吞吐量。该资源接受表示 TCP 缓冲区的字节数的数值。该命令采用以下格式：

`tcp_buffer_size=<空>`

默认值：空

限制：8192 到 65536 字节

注意：将小于 8 KB 的值取整到 8 KB。大于 64 KB 的值设置为 64 KB。

resource_file_path

指定 VNM 资源文件（如 Ether Map）的文件路径。

该命令采用以下格式：

resource_file_path=*./CsResource*

./CsResource

（可选）指定 VNM 资源文件的文件路径。此参数可以留空。

wait_active

确定一加载所有模型服务器就接受连接，还是等到所有模型都处于活动状态服务器才接受连接。如果设置为 *Yes*，则控制面板消息显示 SpectroSERVER 启动期间激活的模型的运行百分比。

该命令采用以下格式：

wait_active=*no*

默认值：no。

max_bind_retry_count

指定侦听套接字绑定的最大重试次数。

该命令采用以下格式：

max_bind_retry_count=*50*

默认值：50。

bind_retry_interval

指定侦听套接字绑定重试之间的延迟（以秒为单位）。

该命令采用以下格式：

bind_retry_interval=*30*

默认值：30 秒。

min_client_version

指定 SpectroSERVER 对客户端连接接受的最低参数块版本。

该命令采用以下格式：

```
min_client_version=0
```

默认值：对于客户端连接最低为 0。

max_connections

指定 SpectroSERVER 接受的最大连接数。连接可以来自客户端，或者来自其他服务器，如另一个 SpectroSERVER、Archive Manager 或位置服务器（主要和本地）。

该命令采用以下格式：

```
max_connections=50 gensv000316 docs002958
```

默认值：50 个连接

超过最大客户端连接数时，连接可能会失败。增加最大 SpectroSERVER 连接数时，有时需要增加工作站上打开的文件描述符数以提高性能。

handshake_timeout

设置连接握手期间交换初始 ID 信息的时间（以秒为单位）。

该命令采用以下格式：

```
handshake_timeout=40
```

默认值：40 秒

vnm_message_timeout

vnm_message_timeout 资源设置在 VNM 之间发送的消息的超时（以秒为单位）。

该命令采用以下格式：

```
vnm_message_timeout=180
```

默认值：180 秒

vnm_close_timeout

vnm_close_timeout 资源设置关闭 VNM 之间的连接的超时（以秒为单位）。

该命令采用以下格式：

```
vnm_close_timeout=180
```

默认值：180 秒

connect_time_limit

设置连接到 VNM 的超时（以毫秒为单位）。

该命令采用以下格式：

```
connect_time_limit=5000
```

默认值：5000 毫秒

rcpd_comm_port

指定远程复制进程后台进程 (rcpd) 的侦听端口。此资源用于容错的数据库备份。

该命令采用以下格式：

```
rcpd_comm_port=0xCAFE
```

默认值：0xCAFE

procd_comm_port

指定 VNM 用于连接到进程后台进程的端口。

该命令采用以下格式：

```
procd_comm_port=0xFEEB
```

默认值：0xFEEB

snmp_trap_port

指定 VNM 通过其接收陷阱的端口。

该命令采用以下格式：

```
snmp_trap_port=162
```

默认值：162。

enable_traps_for_pingables

指定在可 ping 的模型上 CA Spectrum 是否可以接收 SNMP 陷阱。默认情况下，.vnmrc 文件不包含此资源。请使用以下语法，手动将它添加到文件：

```
enable_traps_for_pingables = TRUE
```

默认值：TRUE

device_limit

指定在这样的限制适用时可以建模的最大设备数。该值源自安装期间输入的 CA Spectrum 密钥。此值无法编辑。不是简单主机、可 ping 的设备或节点的代理程序的任何托管节点都是已建模的设备。如果没有适用于 CA Spectrum 安装的设备限制，则没有与此资源关联的值。（另请参阅 max_device。）

该命令采用以下格式：

`device_limit=<`*设备模型数*`>`

设备模型数

指定从 IP 地址 (0x000102bc) 模型类型和设备 (0x0001004b) 模型类型派生的任何模型。

此外，还包括源自 DIRIG_NODE 应用程序的模型和以下宽带模型：MOTCBLMODEM、SAEXPLORER 和 DOCSISCM。

此规则的例外如下：

GEN_HOST；PINGABLE；PATROL_PET 和 WINDOWS_PC。

max_device

适用于其 device_limit 资源有效的 CA Spectrum 产品。默认情况下，模型数达到指定设备限制的 80% 时，CA Spectrum 生成黄色警报。模型数达到指定设备限制的 100% 时，CA Spectrum 生成红色警报。此资源允许您设置一个降至实际设备限制以下的值，从而为您提供早期警告。

该命令采用以下格式：

`max_devices=<`*设备模型数*`>`

设备模型数

请参考其 IP 地址源自设备模型类型且其指定数降至 device_limit 资源值以下的模型。默认值为空。

disable_redundancy_when_using_loopback

在通过环回对设备建模时禁用冗余 (0x11d2c)。仅当 VNM 模型的 use_loopback (0x12bb) 属性设置为 True 时，此参数才适用。

该命令采用以下格式：

`disable_redundancy_when_using_loopback=`*False*

默认情况下，此资源不出现在 .vnmrc 文件中，因此自动计算为 False。

注意：有关详细信息，请参阅*《IT 基础架构建模与管理 - 管理员指南》*。

persistent_alarms_active

防止在 SpectroSERVER 关闭时 CA Spectrum 保留与警报相关的信息。

该命令采用以下格式：

`persistent_alarms_active=<False>`

默认情况下，此资源不出现在 .vnmrc 文件中。先关闭再重新启动 SpectroSERVER 时，CA Spectrum 自动保留与警报相关的信息（如故障排除者分配或状态）。会保留在关闭之前存在的警报。这些警报被视为“持久性”警报。将 persistent_alarms_active=False 添加到 .vnmrc 文件可防止在 SpectroSERVER 关闭时 CA Spectrum 保留与警报-相关的信息。

注意： 我们强烈建议您不要添加 persistent_alarms_active 资源，除非集成了保留与警报相关信息的第三方应用程序。

添加此资源具有以下影响：

- 警报状态新增内容会丢失（仅作为过去的事件记录）
- 在启动时将用新时间戳重新生成 SpectroSERVER 关闭时的现有警报。
- 重新生成的警报将被转发到警报通知工具。

unsupported_attr_poll_interval

指定 CA Spectrum 等待轮询已返回“noSuchName”错误的外部属性的时间。如果在 .vnmrc 文件中未指定此参数，则使用默认值（12 小时）。

该命令采用以下格式：

`unsupported_attr_poll_interval=43200`

默认值：43200 秒（12 小时）

详细信息：

远程复制进程后台进程 (rcpd) 端口号配置 (p. 59)

事件存档 (.vnmrc) 资源

事件存档 (.vnmrc) 资源控制 SpectroSERVER 将事件发送到 Archive Manager 以便记录的过程。以下列表介绍了事件存档 (.vnmrc) 资源：

max_event_records

指定可以在事件日志数据库中存储的最大记录数。

该命令采用以下格式：

`max_event_records=`*记录数*

记录数

指示事件日志记录数。最小值等于 event_record_increment 值。系统存储容量会限制最大值。

默认值：20,000

event_record_increment

指定在记录数超过 max_event_records 值时要从事件日志数据库中删除的记录数。

该命令采用以下格式：

`event_record_increment=`*记录数*

记录数

指示从数据库中删除的事件日志记录数。最小值为 100。

event_batch_max_size

设置每批的最大事件数。如果批已满，则事件批将立即发送到 Archive Manager。

该命令采用以下格式：

`event_batch_max_size=`*1000*

默认值：1000（最大事件数）

event_batch_timeout

确定事件批发送到 Archive Manager 时的时间（以秒为单位）。

该命令采用以下格式：

`event_batch_timeout=`*1*

默认值：*1* 秒

log_user_events

控制是否为每个用户启动的对模型属性值的写入生成事件。值为 True 会导致 VNM 生成事件。

该命令采用以下格式：

log_user_events=*False*

默认值： False

use_log_queue

在生成事件时，将其放置在单独的队列中。这些事件发送到单独线程上的 Archive Manager。如果设置为“TRUE”，则事件发送到与生成它相同的线程上的 Archive Manager。此情况可能会延迟警报创建。

该命令采用以下格式：

use_log_queue=<空>

默认值： <空>

工作线程 (.vnmrc) 资源

SpectroSERVER 是一个多线程进程。在正常操作期间，每个子系统都分配许多工作线程。

SpectroSERVER 维护随时间重新使用的工作线程池。在活动增加期间，子系统使用池中的线程，直至达到各自的限制。工作线程的公共池耗尽时，会创建新的工作线程。池将增大以适应增加的活动。

子系统不再需要的工作线程将返回到公共池供以后使用。从池中提取线程以满足后续要求，或者分配新线程（如果池为空）。在指定的时间内一直不用的线程将从池中删除，而且其资源会返回到系统。此过程称为“老化”。

以下列表介绍了工作线程 (.vnmrc) 资源：

max_total_work_threads

指定可以为所有 SpectroSERVER 子系统分配的最大工作线程数。每个工作线程都占用处理资源并需要相当大的内存块。基于系统容量（内存大小和速度）来设置其值。

此资源的值过高时，SpectroSERVER 可以定期用尽内存。此资源的值过低时，SpectroSERVER 操作可能会变得很慢。

如果最近设置或更改了此资源的值，则在重新启动 SpectroSERVER 时 CA Spectrum 将读取新值。读取值后，它将用于更新 VNM 模型上 WorkThreadsMaxAvail 属性的值。更新 WorkThreadsMaxAvail 的值后，将从 .vnmrc 文件中删除 max_total_work_threads 的值。

该命令采用以下格式：

`max_total_work_threads=`*线程数*

VNM WorkThreadsMaxAvail 属性的默认值为 500。

work_thread_age

子系统不再需要的工作线程将返回到工作线程池。此资源指定工作线程在不被使用的情况下可以保留在池中的时间（以秒为单位）。

将此资源设置为与子系统活动兼容的值。创建工作线程需要大量的处理开销。将此资源的值设置得过低会加重系统资源的负担，原因是为响应工作线程需求而太频繁地创建新线程。将该值设置得过高意味着资源不必要地保持已提交状态。

该命令采用以下格式：

`work_thread_age =`*seconds*

默认值：60 秒

容错警报服务 (.vnmrc) 资源

在容错环境中，警报必须在主要和备用 SpectroSERVER 之间进行同步。服务器进行连接以交换警报信息。如果初始连接尝试失败，则可以进行后续尝试。容错警报服务使用以下设置控制警报同步：

ftasv_enabled

启用警报同步。在主要和备用服务器的 .vnmrc 文件中都包括此命令。

该命令采用以下格式：

`ftasv_enabled=`*true*

默认值：True

重要说明！不能对一个服务器将 ftasv_enabled 设置为 True，而对另一个服务器设置为 False。将相同的值用于这两个服务器。

ftasv_max_conn_retry_count

指定在一次连接尝试失败之后主要服务器尝试连接到备用服务器以便同步的次数。此参数在主要 SpectroSERVER 的 .vnmrc 文件中是必需的。

注意：尝试连接到备用服务器时，主要服务器使用此参数。备用服务器尝试连接到主要服务器时，不使用它。

该命令采用以下格式：

`ftasv_max_conn_retry_count=`*重试次数*

默认值：4 次重试（与备用服务器共进行 5 次同步尝试）。

ftasv_conn_retry_interval

指定主要服务器与备用服务器同步的两次尝试之间的秒数。在主要 SpectroSERVER 的 .vnmrc 文件中使用此参数。

注意：尝试连接到备用服务器时，主要服务器使用此参数。备用服务器尝试连接到主要服务器时，不使用此参数。

该命令采用以下格式：

`ftasv_conn_retry_interval=`*秒数*

默认值：30 秒

ftasv_debug

为警报同步活动启用调试输出。在主要和备用服务器的 .vnmrc 文件中都包括此命令。调试输出将写入每个服务器的 VNM.OUT 文件。每条消息都以“容错警报服务”开头。

该命令采用以下格式：

`ftasvdebug=`*true*

默认值：False

重要说明！如果在主要服务器尝试连接到备用 SpectroSERVER 时后者未运行，则主要服务器会耗尽其同步尝试。因此，将延迟主要 SpectroSERVER 的启动。使用 ftasv_max_conn_retry_count 和 ftasv_conn_retry_interval 的默认设置时，延迟为 2 分钟（4 次重试，间隔为 30 秒）。此延迟是不可避免的，因为它发生在模型激活之前。解决方法是在主要服务器启动时验证备用 SpectroSERVER 是否正在运行。也可以减少重试计数或间隔大小以减少潜在的延迟。

详细信息：

容错 (p. 61)
SpectroSERVER 警报同步 (p. 64)

如何打包 CA Spectrum 实用工具并将其移动到其他计算机

packtool.pl 脚本打包 CA Spectrum 命令行界面 (CLI)、CA Spectrum SS Logger、AlarmNotifier、modelinggateway 工具和 sbgwimport 工具，以便可以将其传输到其他计算机。在移动文件时，脚本保留实用工具的相对目录结构及其支持文件。

注意：这两台计算机必须运行相同的操作系统。

在运行 packtool.pl 脚本之前，请考虑以下要求：

- 您必须是 Solaris 和 Linux 平台上的 root 用户，或者是 Windows 平台上具有管理权限的用户，才能运行 packtool.pl 脚本。
- 将 CA Spectrum 实用工具传输到仅安装了 OneClick 服务器的计算机时，以下要求适用：
  - 每台计算机上 OneClick 服务器的版本和修补程序级别必须匹配。在安装每个修补程序之后，重新运行 packtool.pl 脚本，并重新传输实用工具。
  - 将实用工具传输到的计算机上的安装用户必须与打包文件的计算机上的安装用户匹配。
  - 如果在目标计算机上安装了调试修补程序或 PTF，请在文件传输之后重新安装调试修补程序或 PTF。
- 在打包实用工具的计算机上停止所有 CA Spectrum 进程。
- 在提取 tools_bundle 文件之前，在将实用工具传输到的计算机上停止所有 CA Spectrum 进程。
- 要提取 tools_bundle 文件，您必须是 Solaris 和 Linux 平台上的 root 用户，或者 Windows 上具有管理权限的用户。

重要说明！要运行 CA Spectrum 实用工具，您必须以有效的 CA Spectrum 用户身份登录到目标计算机，否则会发生错误。例如，如果您以“root”身份登录到 Linux 工作站，但在 CA Spectrum 中没有匹配的用户模型，则将收到错误，指出“NO_USER”。

要打包 CA Spectrum 实用工具并将其传输到其他计算机，请采取以下步骤：

1. 使用 packtool.pl 脚本打包 SPECTRUM 实用工具。 (p. 24)
2. 在其他计算机上提取 tools_bundle 文件 (p. 24)。

打包 CA Spectrum 实用工具

在将 CLI、SSLogger、AlarmNotifier、modelinggateway 工具和 sbgwimport 工具移动到其他计算机之前，请运行脚本以绑定它们。

遵循这些步骤:

1. 在打包实用工具的计算机上，验证环境变量 <*$SPECROOT*> 是否设置为 CA Spectrum 安装目录路径。
2. 运行 packtool.pl 脚本，该脚本将打包实用工具及其支持文件。packtool.pl 脚本可在 <*$SPECROOT*>/SS-Tools 目录中找到。

   要从 Bash shell 或其他 UNIX shell 运行脚本，请输入以下命令：

   ```
   <$SPECROOT>/SS-Tools/packtool.pl [-no_notifier | -no_event_alarms]
   [-f file_name]
   ```

   <*$SPECROOT*>

   是在 SpectroSERVER 上安装 CA Spectrum 的目录结构。

   -no_notifier

   指定不希望打包 AlarmNotifier。

   -no_event_alarms

   指定不希望打包 AlarmNotifier EvFormat 或 PCause 文件。

   -f *file_name*

   指定可执行文件的名称。

   将创建一个名为 linux_tools_bundle (Linux)、solaris_tools_bundle (Solaris) 或 nt_tools_bundle.exe (Windows) 的可执行文件，其中包含 CA Spectrum 实用工具及其支持文件。

将 CA Spectrum 实用工具移动到其他计算机

可以将 CA Spectrum 实用工具移动到仅运行 OneClick 服务器的计算机。

遵循这些步骤:

1. 打包 CA Spectrum 实用工具 (p. 24)。
2. 在要提取 CA Spectrum 实用工具的计算机上，将目录转到 <*$SPECROOT*>。
3. 使用二进制模式，通过 FTP 将名为 linux_tools_bundle (Linux)、solaris_tools_bundle (Solaris) 或 nt_tools_bundle.exe (Windows) 的可执行文件传输到当前目录。

4. 从 DOS、Bash 或其他 UNIX shell 提取 tools_bundle 文件。

   CA Spectrum 实用工具及其支持文件将解包到相应的目录结构中。

5. 如果传输了 AlarmNotifier 以及事件和 pcause 文件，请验证 <*$SPECROOT*>/SG-Support/CsResource/preferences/*.prf 中的路径是否具有事件和 pcause 文件的正确位置。此外，请验证 ui= 和 lhandle= 选项。

6. 将 <*$SPECPATH*> 环境变量设置为从其提取 tools_bundle 文件的目录的路径：

   - 在 Windows 上，以变量格式的值创建系统环境变量 <*$SPECPATH*>：

     ```
     driveletter:\PATH_TO_SPECTRUM
     ```

   - 在 Solaris 上，将以下行添加到 /opt/SPECTRUM/spectrum60.env 文件：

     ```
     SPECPATH=PATH_TO_SPECTRUM
     ```

   - 在 Linux 上，将以下行添加到 /opt/SPECTRUM/spectrum80.env 文件：

     ```
     SPECPATH=PATH_TO_SPECTRUM
     ```

7. 在 Solaris 和 Linux 平台上，在 /opt/SPECTRUM 目录的 spectrum*.env 文件中将 BES_LIC_DIR 定义为 PATH_TO_SPECTRUM/bin/VBNS/license。

8. 在 Linux 平台上，将 /usr/bin/perl 复制为 <*$SPECROOT*>/bin。

9. 验证 .hostrc 文件是否包含本地主机名以及从其传输实用工具的计算机的名称。

10. 验证主要位置服务器上的 .hostrc 文件是否包含从其提取 tools_bundle 文件的计算机的主机名。

11. 验证 .LocalRegFile 是否包含正确的主要位置服务器。

12. 将 shell 中的 CLISESSID 设置为使用 CLI 实用工具：

    Windows：

    ```
    设置 CLISESSID=<NUMBER>
    ```

Solaris 和 Linux：

```
导出 CLISESSID=<NUMBER>
```

NUMBER

指示 shell 的任何唯一编号。

13. 验证 Notifier/.alarmrc 是否具有指向 SetScript、ClearScript 和 UpdateScript 的正确路径。这些脚本位于从其提取 tools_bundle 文件的 Notifier 目录中。
14. 重新启动 processd。

    现在可以在此计算机上运行实用工具。

也可以将 CA Spectrum 实用工具移动到未运行 CA Spectrum 的计算机。

遵循这些步骤:

1. 打包 CA Spectrum 实用工具。
2. 在要提取 CA Spectrum 实用工具的计算机上,创建一个目录以解包 CA Spectrum 实用工具及其支持文件。例如，创建以下目录：

    Windows：

    ```
    c:\win32app\spectrum
    ```

    Solaris 和 Linux：

    ```
    /usr/spectrum
    ```

    注意：请将工作目录 (chdir) 转到这些目录。

3. 使用二进制模式，通过 FTP 将可执行文件发送到当前目录。该文件具有以下名称之一：linux_tools_bundle (Linux)、solaris_tools_bundle (Solaris) 或 nt_tools_bundle.exe (Windows)。
4. 从 DOS、Bash 或其他 UNIX shell 执行 tools_bundle 文件。

    CA Spectrum 实用工具及其支持文件将解包到相应的目录结构中。

5. 验证 PATH 变量是否如下所示：

    Windows：

    验证 <*$SPECROOT*>/lib 是否在 PATH 变量中。

    Solaris 和 Linux：

    - 创建 /opt/SPECTRUM 目录。
    - 创建从 <*$SPECROOT*>/lib 到 /opt/SPECTRUM/lib 的链接。
    - 创建从 <*$SPECROOT*>/bin 到 /opt/SPECTRUM/bin 的链接。

6. 将 <*$SPECROOT*> 和 <*$SPECPATH*> 环境变量设置为从其提取 tools_bundle 文件的目录的路径：

   - 在 Windows 上，以变量格式的值创建系统环境变量 <*$SPECROOT*> 和 <*$SPECPATH*>：

     ```
     driveletter:/PATH_TO_SPECTRUM
     driveletter:\PATH_TO_SPECTRUM
     ```

   - 在 Solaris 上，创建 /opt/SPECTRUM/spectrum60.env 并添加以下行：

     ```
     SPECROOT=PATH_TO_SPECTRUM
     SPECPATH=PATH_TO_SPECTRUM
     ```

   - 在 Linux 上，创建 /opt/SPECTRUM/spectrum80.env 并添加以下行：

     ```
     SPECROOT=PATH_TO_SPECTRUM
     SPECPATH=PATH_TO_SPECTRUM
     ```

7. 如果传输了 AlarmNotifier 以及事件和 pcause 文件，请验证 <*$SPECROOT*>/SG-Support/CsResource/preferences/*.prf 中的路径是否具有事件和 pcause 文件的正确位置。此外，请验证 ui= 和 lhandle= 选项。

8. 在 Solaris 和 Linux 平台上，在 /opt/SPECTRUM 目录的 spectrum*.env 文件中将 BES_LIC_DIR 定义为 PATH_TO_spectrum/bin/VBNS/license。

9. 验证 .hostrc 文件是否包含本地计算机主机名以及从其传输实用工具的计算机的名称。

10. 验证 .LocalRegFile 是否包含正确的主要位置服务器。

11. 验证 vnmsh/.vnmshrc 是否包含正确的主要 SpectroSERVER 名称。

12. 安装 processd 服务：

    Windows：

    - 如果未安装 SRAdmin，请安装它，如下所示：

      ```
      shell> cd %SPECROOT%\Install-Tools\sdic\nt
      shell> .\sradmin --install
      shell> .\sradmin --start
      ```

    - 安装 processd 服务：

      ```
      shell> cd %SPECROOT%\lib\SDPM
      shell> .\processd.exe --install --username USERNAME --password PASSWORD
      shell> .\processd.exe --start
      ```

      注意： 如果 processd 未启动，请重新启动计算机。

Solaris 和 Linux：

- 将 <*$SPECROOT*>/lib/SDPM/processd_init.sh 复制到 /etc/init.d/processd。
- 将 <*$SPECROOT*>/lib/SDPM/processd.pl 复制到 /etc/init.d。
- 创建从 /etc/init.d/processd 到 /etc/rc2.d/S99processd 的链接。
- 使用 /etc/init.d/processd start 来启动 processd。

13. 验证主要位置服务器上的 .hostrc 文件是否包含从其提取 tools_bundle 文件的计算机的主机名。
14. 将 shell 中的 CLISESSID 设置为使用 CLI 实用工具：

    Windows：

    ```
    设置 CLISESSID=<NUMBER>
    ```

    Solaris 和 Linux：

    ```
    导出 CLISESSID=<NUMBER>
    ```

    NUMBER

    指示 shell 的任何唯一编号。

15. 验证 Notifier/.alarmrc 是否具有指向 SetScript、ClearScript 和 UpdateScript 的正确路径。这些脚本位于从其提取 tools_bundle 文件的 Notifier 目录中。
16. 在 Windows 平台上，如果要使用 AlarmNotifier 脚本，请从 http://www.cygwin.com 安装 Cygwin。请确保 PATH 变量中包含 Cygwin bin 目录。

    现在可以在此计算机上运行实用工具。

详细信息：

打包 CA Spectrum 实用工具 (p. 24)

第 2 章： 设置分布式 SpectroSERVER 环境

此部分包含以下主题：

名称解析要求

SpectroSERVER 系统或 OneClick Web 服务器系统必须能够将 SpectroSERVER 的主机名解析为 IP 地址。

我们建议使用 hosts 文件进行名称解析。此做法可确保网络停机不会影响 SpectroSERVER 名称解析。

CA Spectrum 和多个接口

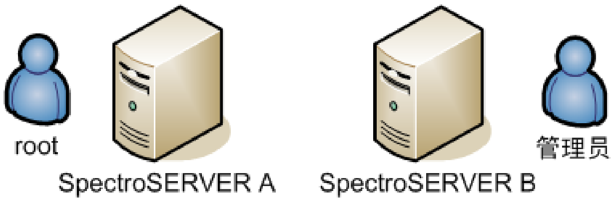
所有 CA Spectrum 服务器都绑定到所有可用的接口并侦听它们，而且用未完全限定的主机名通告自身。在配置管理拓扑时，可达到灵活的效果。为了建立连接，CA Spectrum 组件服务器将按操作系统确定的顺序尝试所有接口，直到连接成功为止。

SpectroSERVER 之间的通信

使用同一用户帐户在分布式环境中安装并运行 SpectroSERVER。因此不需要进一步的用户模型配置。以不同用户身份安装并运行的 SpectroSERVER 无法进行通信。

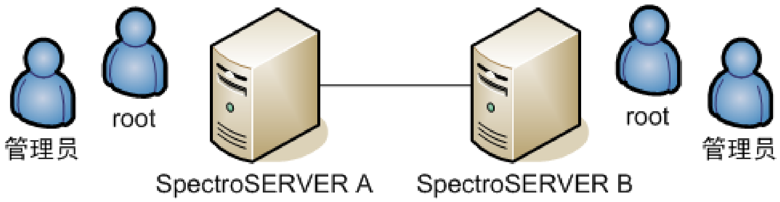
示例：SpectroSERVER A 和 B 无法进行通信

在此示例中，SpectroSERVER A 以“root”用户身份运行。SpectroSERVER B 以管理员身份运行。它们无法建立连接，因为请求无法通过服务器安全性：



示例：SpectroSERVER 到 SpectroSERVER 连接的配置

此示例显示 SpectroSERVER B 上以“root”用户身份登录的用户和 SpectroSERVER A 上以“管理员”身份登录的用户。此配置使这两个 SpectroSERVER 可以进行通信：



端口冲突解决方案

为避免与其他应用程序发生端口冲突，可以更改 CA Spectrum 进程和服务使用的默认端口号。

下表介绍了 CA Spectrum 进程和服务的默认端口。

注意：在适用的情况下，此表中的端口注有“不用于远程连接”。CA Spectrum 不用于远程连接的端口通常不涉及防火墙配置。

端口（十进制）	端口（十六进制）	**CA Spectrum 组件**	**协议**	**注意**
80	0x0050	OneClick/SRM	HTTP	无
162	0x00A2	SpectroSERVER	UDP	用于侦听 SNMP 陷阱的端口。
3306	0x0CEA	SRM 数据的 mysql ArchMgr (Archive Manager)	TCP/ODBC/JDBC	仅当将 SRM 数据库从 Windows 迁移到 Linux 或 Solaris 时，才用于远程连接。
3307	0x0CEB	Solaris/Linux 上 BOXI 的 mysql	TCP	不用于远程连接。
6844	0x1ABC	SDC	TCP	无
6400	0x1900	BOXI CMS	TCP	不用于远程连接。
7777	0x1E61	CLI vnmshd	TCP	请参阅 CLI 后台进程 (vnmshd) (p. 59)。
14001	0x36B1	OneClick 服务器	TCP/CORBA	请参阅 OneClick Web 服务器主机 (p. 55)。
14002	0x36B2	SpectroSERVER	TCP/CORBA	请参阅 SpectroSERVER (p. 57)。
14003	0x36B3	ArchMgr	TCP/CORBA	请参阅 Archive Manager (p. 57)。
14004	0x36B4	LocServ	TCP/CORBA	请参阅位置服务器 (p. 58)。
14006	0x36B6	命名服务	TCP/CORBA	无
31415	0x7AB7	通过 SpectroSERVER 的 Telnet		无
46517	0xB5B5	sradmin	TCP	远程管理后台进程。 **注意：**有关 sradmin 进程的详细信息，请参阅*《安装指南》*。
47870	0xBAFE	ArchMgr	TCP/SSAPI	请参阅 Archive Manager (p. 57)。
48879	0xBEEF	SpectroSERVER	TCP/SSAPI	请参阅 SpectroSERVER (p. 57)。

端口（十进制）	端口（十六进制）	CA Spectrum 组件	协议	注意
51966	0xCAFE	rcpd	TCP	请参阅远程复制进程后台进程 (rcpd) (p. 59)。
56063	0xDAFF	LocServ	TCP	请参阅位置服务器 (p. 58)。
61904	0xF1D0	processd	TCP	此端口不可配置。
64222	0xFADE	TL1d	TCP/EPI	TL1 网关代理
65259	0xFEEB	processd	TCP	此端口不可配置。

DSS 环境要求

分布式环境必须满足以下条件，才能使多个 SpectroSERVER 和多个格局的建模可以表示这些服务器：

- 每个格局在数据库及其关系中都必须包含模型类型的相同建模目录。此复制为 CA Spectrum 智能提供了一致性基础。
- 每个格局都必须包含相同的用户模型，您可以在 OneClick“用户”选项卡中查看这些模型。
- 将在不同格局中建模的子网之间的连接数减到最少。
- 限制在多个格局中建模的相同设备的数量。
- 将唯一的格局句柄分配给每个建模的格局。然后，CA Spectrum 可以将它与 DSS 环境中的其他格局区分开。

详细信息：

分配格局句柄 (p. 36)

位置服务器

在安装 SpectroSERVER 时，还会自动安装*位置服务器*。此服务器可标识和找到网络上的其他 CA Spectrum 服务。CA Spectrum 进程使用位置服务器来确定这些服务正在运行的位置。Processd 可启动和停止位置服务器。

在分布式环境中，CA Spectrum 使用位置服务器维护 SpectroSERVER 格映射并为客户端应用程序提供连接服务。在 CA Spectrum 安装期间，可以将格局映射（或 CA Spectrum 域）中的一个位置服务器指定为*主要位置服务器 (MLS)*。

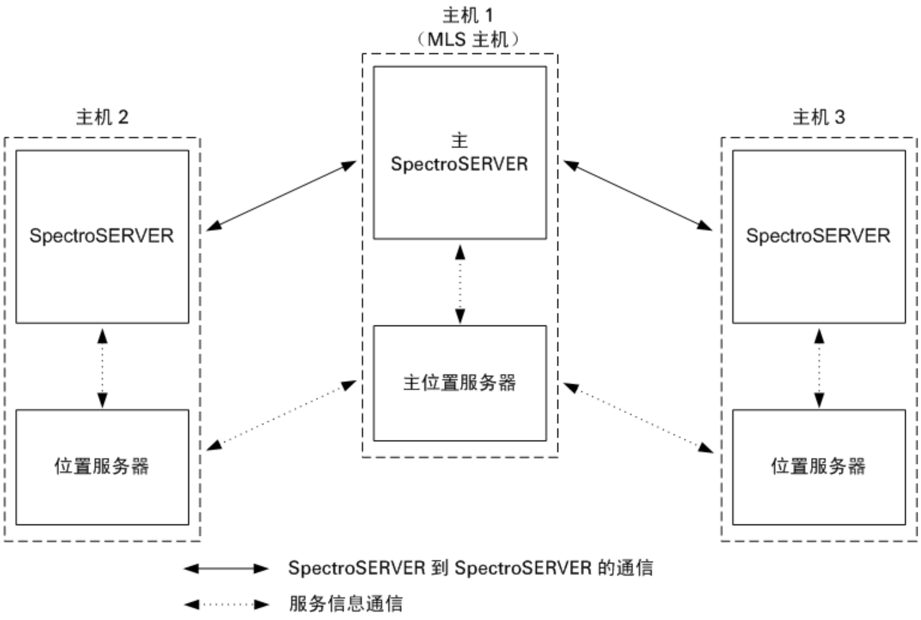
在高度可靠的计算机上安装该主要位置服务器。此服务器在位置服务器之间传播服务通告，并在位置服务器之间传达服务位置。

在下图中，主机 1 上的位置服务器已配置为此环境的主要位置服务器。由于主要位置服务器驻留在与 SpectroSERVER 相同的主机上，因此该 SpectroSERVER 被视为主要 SpectroSERVER。

所有服务器到服务器的通信都通过主要位置服务器主机进行路由。两个非 MLS SpectroSERVER 只能通过主要 SpectroSERVER 进行通信。

图表中主机 2 上的 SpectroSERVER 具有分布式服务及在主机 3 的 SpectroSERVER 上建模的资源。有关远程资源的所有通信都通过主要位置服务器主机进行路由。主机 2 和主机 3 上的服务器不直接进行通信。

重要说明！此图表反映了由一个 MLS 和多个子 SpectroSERVER 组成的两层设置。这样的配置是建议的标准配置。虽然可以由具有分层关联关系（多层部署）的多个 MLS 组成更复杂的配置，但是建议不要使用它们。多层部署增加了复杂性，却未提供更多的功能。



位置服务器的交互方式

每个 CA Spectrum 安装包括以下两个文件：

- .LocalRegFile 允许 CA Spectrum 进程查找其位置服务器。
- LS/.locrc 是在本地安装的位置服务器的配置文件。

这些文件管理位置服务器的层次结构以及每个位置服务器与主要位置服务器之间的交互。

在分布式 SpectroSERVER 部署中，位置服务器进行如下交互：

1. SSAPI 应用程序和服务器读取 .LocalRegFile 以确定要使用的位置服务器。
2. 每个本地位置服务器读取 /LS/.locrc 文件以确定哪个服务器是主要位置服务器。
3. 本地位置服务器连接到主要位置服务器以查找网络服务。如果主要位置服务器变得不可用，则其他位置服务器将服务信息保留在内存中，直到主要位置服务器可用为止。

主要位置服务器连接

每个 SpectroSERVER 都必须具有与在指定为主要位置服务器的服务器上运行的 SpectroSERVER 的连接。此要求在 <*$SPECROOT*>/LS/.locrc 文件中指定。

与主要位置服务器的连接需要以下配置：

- 每个 SpectroSERVER 都必须允许其他 SpectroSERVER 主机与之连接。连接要求在每个服务器上的 .hostrc 文件中有相应的条目。
- 每个 SpectroSERVER 都需要运行其他 SpectroSERVER 所用的用户帐户的 CA Spectrum 用户模型。

注意： 主要格局（或 .locrc 文件中 MAIN_LOCATION_HOST_NAME 的值）在 CA Spectrum 中建模。每个 SpectroSERVER 的 VNM 拓扑都包含此格局，除非它已在其他位置建模。如果与主要格局的连接丢失，则会在此模型上生成警报。

指定新的主要位置服务器

可以更改指定为主要位置服务器的计算机。

在 SpectroSERVER 上，通过“位置服务器配置”对话框或通过 .locrc 文件指定新的主要位置服务器。

在 OneClick Web 服务器上，在“OneClick 管理”页的“CA Spectrum 配置”页中设置主要位置服务器信息。此操作将更新 OneClick Web 服务器连接到的主要位置服务器的名称。

注意：如果在 SpectroSERVER 上更改主要位置服务器而不更新 OneClick Web 服务器，则 CA Spectrum 无法正常工作。有关详细信息，请参阅*《管理员指南》*。

示例：在 SpectroSERVER 上指定新的主要位置服务器

分布式网络由计算机 1 到 4 组成，其中计算机 1 被指定为主要位置服务器。要使计算机 4 成为主要位置服务器，需要先将计算机 4 重新配置为新的主要位置服务器。然后，将计算机 1、2 和 3 重新配置为指向作为主要位置服务器的计算机 4。

遵循这些步骤:

1. 启动 SpectroSERVER 控制面板。
2. 从“配置”菜单中选择“位置服务器”。

   将打开“位置服务器配置”对话框。
3. 将“位置服务器特征”区域中的“主要 LS 主机”字段更改为指向计算机 4。

注意：也可以编辑 <*$SPECROOT*>/LS 目录中的 .locrc 文件来更改主要位置服务器。.locrc 文件中的以下条目标识主要位置服务器：

MAIN_LOCATION_HOST_NAME=*Computer 4*

详细信息：

建立容错 (p. 68)

格局映射完整性保护

CA Spectrum 用户通常维护单独的生产和测试环境。通过防止测试环境中的 SpectroSERVER 连接到生产环境 SpectroSERVER 中作为其主要位置服务器的 SpectroSERVER（或者相反）来保护格局映射的完整性。

格局句柄

分布式部署中的 SpectroSERVER 按*格局句柄*标识每个格局。每个格局必须具有唯一的格局句柄，它是一个可被 4 整除且范围在 4 到 16376 之间的数字。（在十六进制格式中，句柄范围在 0x100000 到 0xffe00000 之间，其中较低的 20 位设置为零。）已编码的格局句柄显示在与该格局关联的所有视图的顶部。

注意： 我们建议将连续的编号方案用于格局句柄。可以将格局句柄与重要的格局功能（如建筑物编号或子网 IP 地址的某个部分）关联。但是，会将您的条目编码到格局句柄的 12 个最高有效位中。因此，结果可能看起来与相应的格局功能无关。

分配格局句柄

通过安装 CA Spectrum 时出现的“SpectroSERVER 验证”对话框或者通过使用名为 lh_set 的实用工具，可以分配格局句柄。

重要说明！请在首次运行 SpectroSERVER 之前运行 lh_set 实用工具。否则，CA Spectrum 将分配默认的格局句柄，这样 CA Spectrum 每次分配的格局句柄都相同。因此，在配置多个格局时，可以创建重复的格局句柄。永远无法从同一应用程序同时访问这些格局。

遵循这些步骤:

1. 导航到 SS 目录。
2. 输入以下命令：

   ```
   ../SS-Tools/lh_set <landscape handle>
   ```

   可以通过十进制或十六进制记数法指定新的格局句柄。如果使用十进制记数法，则 lh_set 实用工具将条目转换为十六进制格局句柄。

注意：有关详细信息，请参阅《*安装指南*》。

详细信息：

DSS 环境要求 (p. 32)

更改格局句柄

如果 SpectroSERVER 已启动，则更改 SpectroSERVER 数据库的格局句柄的过程需要更多步骤。格局句柄包括在 SpectroSERVER 数据库中每个模型的模型句柄中。因此，更改格局句柄需要将所有的模型句柄从旧的格局句柄转换为新的格局句柄。

更改在启动时自动创建的所有模型的格局句柄，并在多个资源文件中更新该格局句柄。

遵循这些步骤:

1. 使用“建模网关”工具从要更改其格局句柄的 SpectroSERVER 中导出模型。
2. 关闭 SpectroSERVER。
3. 使用 SSdbload 实用工具初始化数据库。

   注意：有关详细信息，请参阅*《数据库管理指南》*。
4. 使用 lh_set 实用工具分配新的格局句柄。
5. 启动 SpectroSERVER。
6. 使用“建模网关”工具将模型导入到 SpectroSERVER 中。

注意：有关详细信息，请参阅《Modeling Gateway Toolkit 指南》。

进程后台进程 (processd)

CA Spectrum 使用一个名为“进程后台进程 (processd)”的进程启动和跟踪后台进程，以允许您在 DSS 环境中的多个服务器上控制进程。在应用程序（如 CA Spectrum 控制面板）发出请求时，此后台进程会启动进程。也可以使用安装票单配置 processd 以在系统启动时自动启动进程。安装票单还会自动重新启动意外停止的关键进程。

CA Spectrum 安装程序将 processd 配置为在 Solaris 系统（它必须以 root 用户身份运行）和 Windows 系统（它以 LocalSystem 帐户身份运行）上自动启动。

仅当应用程序请求启动和监控进程的操作时，processd 才会进行这样的操作。通常，processd 在后台运行。

如果进程未启动或者未正常工作，则 processd 会将错误消息写入 <*$SPECROOT*>\lib\SDPM\processd_log 文件。该错误消息包括用于确定问题的信息。

在重新启动时，processd 创建 processd_log.bak 文件以保存旧的错误消息，并将任何新的错误消息附加到 processd_log 文件。

重要说明！在尝试此部分所述的任一过程之前，您必须完全熟悉 CA Spectrum、分布式联网和网络配置。

详细信息：

安装票单文件 (p. 39)

Windows 和 Solaris 环境中的 Processd 差异

由于本地安全体系结构中的差异，processd 后台进程在 Windows 和 Solaris 平台上的工作方式稍有不同。

- 在 Windows 环境中请求从远程连接启动进程时，进程将以在“Windows 服务配置”对话框中指定的用户身份启动。在 Solaris 上，进程以发出请求的用户身份启动。
- 对于在用户注销之后（或者在任何人登录之前）必须保持运行的服务器进程，在 Windows 环境中将使用 SERVERPROCESS 字段。为防止在注销期间 Windows 关闭这些进程，这些进程将以在“Windows 服务配置”对话框中指定的用户身份启动。在 Solaris 上，进程以在安装票单中指定的 TICKETUSER 身份启动。

详细信息：

安装票单文件 (p. 39)

在 Processd 中更改 Windows 密码

在 Windows 上安装期间，系统会提示您在“Windows 服务配置”对话框中输入用户名和密码。processd 后台进程使用此用户名和密码以指定用户的身份启动 CA Spectrum 进程。通过提供安全信息，这些进程可以在没有用户已登录时运行。

注意：有关详细信息，请参阅*《安装指南》*。

如果在“Windows 服务配置”对话框中指定的密码发生更改或过期，则可以用新密码更新 processd。

重要说明！如果密码包含特殊字符（如感叹号 (!)），请用反斜杠 (\) 将这些字符转义。或者，可以从命令提示符更改密码。

遵循这些步骤:

1. 以本地管理员组的成员身份登录。

   您必须是 CA Spectrum 用户组成员和管理员才能更改 processd 用户名和密码。

2. 打开命令提示符。

3. 导航到 /lib/SDPM 目录。

4. 输入以下命令：

   ```
   processd --install --username user --password newpassword
   ```

   user

   指定用户名。

   newpassword

   指定此用户名的新密码。

   如果用户名或密码包含 shell 可能曲解的特殊字符，请用引号将用户名或密码括起来。例如，用户名包含一个域名和用户，且带有反斜杠 (\) 分隔符。密码包含一个星号 (*)。这两个字符都是特殊字符。因此，用引号将用户名和密码括起来，如以下示例所示：

   ```
   processd --install --username “DOMAIN\JSmith” --password “283EJ*”
   ```

   注意： 如果 Windows 用户名的密码发生更改，但是 processd 服务密码未更新，则 processd 不会启动。在 Windows 事件日志中将生成以下事件：

   - 使用当前密码的登录尝试因下列错误而失败：
登录失败：未知的用户名或错误密码。
   - 由于以下错误，CA Spectrum 进程后台进程服务无法启动：
由于登录失败，服务未启动。

安装票单文件

可以将 processd 配置为在系统启动时启动进程。还可以选择在进程停止运行后重新启动它们。称为*安装票单*的文件支持此功能。

安装票单使用以下两个目录中的文件：

- <*Spectrum Installation Directory*>/lib/SDPM/partslist
- <*Spectrum Installation Directory*>/lib/SDPM/runtime

partslist 目录包含单独的安装票单文件。

runtime 目录包含 *<PID>*.rt 形式的编码文件，其中 *<PID>* 是正在运行的进程的进程 ID。

注意：SDPM 代表 CA Spectrum Distributed Process Manager。

可以将安装票单添加到 partslist 目录。新安装票单必须符合特定的格式设置规则。重新启动 processd 以标识 partslist 目录中任何新的或修改的安装票单。在 processd 启动时读取这些文件，并将其存储在缓存内存中以供将来使用。

安装票单文件遵循的命名约定引用它包含其配置信息的进程。安装票单文件名采用 *<PARTNAME>*.idb 形式。*<PARTNAME>* 变量是用于标识 processd 所控制的进程的内部键。

以下定义字段用于安装票单。格式为 *<fieldname>*;*<value>*；其中 *<fieldname>* 是下面显示的名称之一。*<value>* 是一个字符串，提供对应字段名称的定义。不支持引号。

PARTNAME

用不含空格的多字符字符串标识特定的进程/应用程序。此应用程序的安装票单具有 *<PARTNAME>*.idb 形式的文件名。*<PARTNAME>* 是进程名称。

APPNAME

将应用程序的名称定义为多字符字符串。

WORKPATH

指定要运行应用程序的位置。提供此字段的值，*之后*可将其用作稍后所述的 ARGV 字段的一部分。

LOGNAMEPATH

指定从应用程序输出的日志文件。此字段必须以 *<$SPECROOT>* 或 *<$WORKPATH>* 开头。不允许使用空格。

ADMINPRIVS

保留它仅供在 Purism 提供的安装票单中使用。在创建的安装票单中将其注释掉。

AUTORESTART

指示在进程停止后是否自动重新启动它。如果此字段未包括在安装票单中，则默认情况下禁用自动重新启动。接受的值有 Y、y、N、n。

AUTOBOOTSTART

指示每当 processd 启动时是否都启动进程。如果此字段未包括在安装票单中，则默认情况下禁用该功能。接受的值有 Y、y、N、n。

STATEBASED

指示在启动时进程是否具有多个状态。通常，在通信准备就绪时，进程发送“已就绪”票单。保留此字段仅供在 Purism 提供的安装票单中使用。如果此字段未包括在安装票单中，则默认情况下禁用它。接受的值有 Y、y、N、n。

NUMPROCS

指定可以在一个平台上运行的进程实例数。可接受数值。

RETRYTIMEOUT

指定在发生故障之后多少秒 processd 尝试重新启动应用程序。

TICKETUSER

定义在设置了 AUTOBOOTSTART 和/或 AUTORESTART 字段时有权运行进程的用户的用户名。仅当那些字段包括在安装票单中时，才需要此字段。此字段在 Windows 上不适用，因为所有进程都以 processd 安装用户身份运行。

RETRYMAX

指定在指定的 RETRYTIMEOUT 期限内 processd 尝试重新启动应用程序的次数。

STARTPRIORITY

指示该进程相对于其他进程的相对启动优先权。可使用以下值：

- 10，表示独立进程
- 20，表示依赖于独立进程的进程
- 30，表示依赖于 SpectroSERVER 的进程

SERVERPROCESS

向 processd 指示在用户注销之后进程是否继续运行。启用此字段时，进程始终以有权运行 processd 服务的用户身份启动。此字段仅在 Windows 环境中适用。对于以下进程，默认值为“是”：

- Archive Manager (ARCHMGR.idb)
- 位置服务器 (LOCSERV.idb)
- SpectroSERVER (SS.idb)

SERVICE

指示票单是否表示 Windows 服务。允许 processd 从服务控制管理器管理 Windows 服务。

ENV

将一个或多个变量添加到应用程序环境。在单独的行上列出多个值，宏 <CSPATHSEP> 介于值与行尾的分号之间。

ARGV

定义进程的参数列表，其中包括可执行文件路径和任何数量的参数（允许使用空格）。

示例：安装票单

安装票单应用以下语法约定：

- 以磅字符 (#) 开头的任何行都视为注释。
- 分号 (;) 必须跟随所有字段名称和每个字段名称之后的所有值。
- 忽略第二个分号之后的任何内容。
- 可以使用四个宏：<CSEXE>、<CSBAT>、<CSCMD> 和 <CSPATHSEP>。根据平台，替换相应的定义。使用 <CSPATHSEP> 而不是实际的路径分隔符以避免解析冲突。

以下示例显示了一个有效的安装票单：

```
# 用于 SpectroSERVER 后台进程的 Processd 安装票单。
PARTNAME;SS;
APPNAME;SpectroSERVER Daemon;
WORKPATH;$SPECROOT/SS;
LOGNAMEPATH;$WORKPATH/VNM.OUT;
ADMINPRIVS;y;
#AUTORESTART;N;
#AUTOBOOTSTART;N;
STATEBASED;y;
NUMPROCS;3; // unlimited
RETRYTIMEOUT;0; // seconds
#TICKETUSER;$USER;
RETRYMAX;0;     // retries
STARTPRIORITY;20;
SERVERPROCESS;Y;
#ENV;<var>=<value>;
ARGV;$SPECROOT/SS/SpectroSERVER<CSEXE>;
```

以下示例显示了另一个有效的安装票单：

```
# 用于 Visibroker 命名服务的 Processd 安装票单
PARTNAME;NAMINGSERVICE;
APPNAME;Visibroker Naming Service;
WORKPATH;$SPECROOT/bin/VBNS;
LOGNAMEPATH;$WORKPATH/NAMINGSERVICE.OUT;
ADMINPRIVS;y;
AUTORESTART;y;
AUTOBOOTSTART;y;
#STATEBASED;N;
NUMPROCS;1; // one per host
RETRYTIMEOUT;600; // 10 minutes
#TICKETUSER;$USER;
RETRYMAX;5;     // 5 retries
STARTPRIORITY;10;
SERVERPROCESS;Y;
#ENV;<var>=<value>;
ENV;CLASSPATH=$SPECROOT/lib/vbjorb.jar<CSPATHSEP>;
ENV;CLASSPATH=$SPECROOT/lib/vbsec.jar<CSPATHSEP>;
ENV;CLASSPATH=$SPECROOT/lib/lm.jar<CSPATHSEP>;
ENV;CLASSPATH=$SPECROOT/lib/sanct6.jar<CSPATHSEP>;
ARGV;
$SPECROOT/bin/JavaApps/bin/nameserv<CSEXE>
-DORBpropStorage=
$SPECROOT/.corbarc
-Dvbroker.orb.admDir=
$SPECROOT/bin/VBNS
-Dborland.enterprise.licenseDir=
$SPECROOT/bin/VBNS/license
-Dborland.enterprise.licenseDefaultDir=
$SPECROOT/bin/VBNS/license
-Djava.endorsed.dirs=
$SPECROOT/lib/endorsed
-Dorg.omg.CORBA.ORBClass=
com.inprise.vbroker.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=
com.inprise.vbroker.orb.ORB
com.inprise.vbroker.naming.ExtFactory;
```

启动新的安装票单进程

如果添加了安装票单文件，则可以使用重新启动选项启动在票单中指定的进程。使用重新启动选项，无须停止并重新启动 processd 及其所监控的所有进程。

在 Solaris 上按照以下步骤操作：

1. 以 root 用户身份登录。
2. 导航到 /lib/SDPM 目录。
3. 输入以下命令：

   ```
   processd.pl restart
   ```

   将启动进程。

在 Windows 上按照以下步骤操作：

1. 以 CA Spectrum 用户组的成员身份登录。
2. 打开命令提示符。
3. 导航到 /lib/SDPM 目录。
4. 输入以下命令：

   ```
   perl processd.pl restart
   ```

   将启动进程。

停止并重新启动 Processd

如果您怀疑 lib/SDPM/runtime 目录已损坏，请停止并重新启动 processd。

在 Solaris 上遵循这些步骤:

1. 以 root 用户身份登录。
2. 导航到 /lib/SDPM 目录。
3. 输入以下命令：

   ```
   processd.pl stop
   ```

4. 验证所有 CA Spectrum 进程是否均已关闭（否则 SpectroSERVER 可能会出现问题）。
5. 删除 /lib/SDPM/runtime 目录中的所有条目。
6. 通过输入以下命令重新启动 processd：

   ```
   processd.pl start
   ```

在 Windows 上遵循这些步骤:

1. 以 CA Spectrum 用户组的成员身份登录。
2. 打开命令提示符。
3. 导航到 /lib/SDPM 目录。
4. 输入以下命令：

   `perl processd.pl stop（或 start）`

注意： 在 Windows 上，processd.pl <*start/stop*> 命令还会停止和启动作为 NT 服务运行的 CA Spectrum 进程（如 MySQL 和 VisiBroker）。

详细信息：

Solaris 环境中的 SpectroSERVER 关闭 (p. 48)

安装期间 Userconf 进程的工作方式

userconf 进程在后台运行 processd。Userconf 允许您在安装期间和之后，当用户登录到 CA Spectrum 时执行 CA Spectrum 的用户-特定的配置。

userconf 进程在 CA Spectrum 安装期间执行以下任务：

1. 运行 userconf -install %SPECROOT%，会进行以下更改：
   - 将 SPECROOT 和 SPECPATH 添加到系统环境。
   - 将 $SPECPATH\lib 添加到 $PATH 系统变量。
   - 如果安装目录已更改，则从路径中删除旧的 $SPECPATH\lib 条目。
   - 从路径中删除 %SPECPATH%\lib（如果它存在）。
   - 修改注册表，以便每次用户登录时 userconf 都会运行（不带参数）。
2. 运行 userconf -start 以启动 processd。-start 标志启动 processd 而不验证用户是否为 CA Spectrum 用户组的成员。

   注意： 安装程序将当前用户添加到 CA Spectrum 用户组。但是，在用户先注销再登录之后，更改才会生效。添加 -start 标志，以便 processd 启动而不验证用户是否为 CA Spectrum 用户组的成员。然后，安装继续，而不要求用户先注销再登录。
3. 如果用户是首次安装 Exceed（在 CA Spectrum 安装之后发生），则运行 userconf -restart 以停止并重新启动 processd。然后，Processd 可以在 Exceed 安装过程中更新 PATH 环境更改。

用户登录时 Userconf 的工作方式

安装会将 userconf 进程添加到注册表项 Run。在用户登录时，进程将自动启动，并检查用户是否为 CA Spectrum 用户组的成员。userconf 进程应用以下规则：

- 如果用户是 CA Spectrum 用户组的成员，则 userconf 验证 cygwin 和 Exceed，并根据需要在启动 processd 之前重新配置。如果安装了 Microsoft VC++，则 userconf 将对其进行适当配置以用于 CA Spectrum SDK。
- 如果用户不是用户组的成员，则会显示一条消息，指出已安装 CA Spectrum 但是用户不在 CA Spectrum 用户组中。

  为了使用户可以使用 CA Spectrum，请将用户帐户添加到 CA Spectrum 用户组。然后，用户必须先注销再登录。

  不希望使用 CA Spectrum 的用户可以选中“用户不希望运行 Spectrum”复选框。通过此选项，在用户登录时 userconf 继续运行，但是进程以静默方式退出。

  注意：要重新查看消息，请运行以下命令：

  ```
  userconf -install %SPECROOT%
  ```

网络分区

对已配置的网络建摸时，网络分区并不重要。但是，如果创建多个格局并要使用**发现**创建拓扑层次结构，请使用支持发现的方法。

您的目标是将格局中的模型重复和格局之间的连接数减到最少。为了便于故障隔离，请复制一些模型，如连接两个 IP 子网的路由器。

用于网络分区的选项如下：

- 创建作为子网的分区，它们由 IP 地址边界定义。IP 地址范围提供了一种限制发现探索的最便利方法。
- 如果 IP 地址边界未提供分隔格局的方法，请使用唯一的社区名称来区分每个格局中的设备。此方法更为复杂。将这些名称添加到每个设备的社区名称表。
- 根据包含多个 IP 子网地址的其他参数（如时区或状态）创建分区。大多数情况下，必须手动对网络建模。但是，请在格局之间保持尽可能少的连接并遵守建模规则。过多的连接可能会损害 DSS 的优势。

注意：有关详细信息，请参阅 *《IT 基础架构建模与管理 - 管理员指南》*。

分布式环境中的重复模型

在 DSS 环境中，CA Spectrum 通过将模型指定为“主”模型来跟踪重复模型（例如，在多个格局上存在的用户模型）。主模型驻留在包含“根”主要位置服务器 (MLS) 的主机上运行的 SpectroSERVER 上。

重要说明！如果在 DSS 环境中更改根 MLS，则所有重复模型的状态都将更新为与 MLS 上的“主”模型匹配。在任何 DSS 环境中，MLS 都具有最终的决定权。

CA Spectrum 使用主模型同步所有重复模型的信息。向不是主模型的重复模型发出的修改请求将中继到主模型的格局中。然后，主模型将请求分发到所有的其他重复模型。

无法联系主格局错误

无法联系所编辑模型的主格局时，编辑操作可能会失败。OneClick 将报告关系错误，如来自 SpectroSERVER 的以下错误：

操作失败，因为所编辑的模型存在于多个格局中且无法联系其主格局。

此类型的错误可能指示在“根”MLS 上运行的 SpectroSERVER 上的主模型不可访问。在分布式环境中添加或删除重复模型时，将通过主格局路由它们。如果该 SpectroSERVER 已关闭或者无法联系，则关系失败并生成错误。

在以下任一情况下都可能无法联系主格局：

- 主 SpectroSERVER 或者重复模型和主 SpectroSERVER 之间的中间 SpectroSERVER 未在运行。
- 网络问题正在阻止 SpectroSERVER 访问。
- 安全问题正在阻止 SpectroSERVER 通信。

主机资源配置文件 (.hostrc)

.hostrc 文件限制客户端应用程序对每个本地 SpectroSERVER 的访问。客户端应用程序无法连接到本地格局，除非 .hostrc 文件已配置为允许此访问。可以使用文本编辑器编辑 .hostrc 文件。

注意：有关详细信息，请参阅*《管理员指南》*。

默认情况下，.hostrc 文件最初以本地主机名安装，这会限制所有的远程访问。通过添加单个主机名，可以允许与该服务器的双向连接。通过添加“+”号，可以允许与所有远程服务器的双向无限制访问。通过将计算机名添加到 .hostrc 文件，可以允许从单个远程服务器访问客户端。

CA Spectrum 按主机名或按 IP 地址在多个层中实现主机安全性。我们建议在包括与主机名关联的所有 IP 地址的 .hostrc 文件中列出主机名。如果使用 IP 地址进行连接，则由主机名启动的连接尝试并不总是成功。

DSS 环境中的时区

在跨多个时区的 DSS 安装中，在 SpectroSERVER 上发生的所有活动都反映服务器的本地时间，其中包括排定的事件。在 OneClick 中创建并应用于已建模设备的所有排定都将根据 SpectroSERVER 或设备格局的本地时间开始和结束。

注意：有关详细信息，请参阅*《操作员指南》*。

SpectroSERVER 关闭

我们建议您在关闭正在运行 SpectroSERVER 的服务器之前，使用 CA Spectrum 控制面板关闭 SpectroSERVER。但是，如果通过操作系统关闭过程来关闭 SpectroSERVER，会增加 processd 停止的时间。

Solaris 环境中的 SpectroSERVER 关闭

在 Solaris 环境中，默认情况下，processd 等待 20 秒，以便在它停止之前关闭所有子进程。如果 SpectroSERVER 关闭所用时间很长，则 processd 会等待较长的时间再停止。使用 PROCESSD_SHUTDOWN_TIMEOUT 环境变量可更改默认值。将此变量及其相应值（以毫秒为单位）添加到 <*$SPECROOT*>/spectrum60.env 文件。

例如，要使 processd 等待 60 秒以便关闭所有子进程，请将以下行添加到 spectrum60.env 文件：

```
PROCESSD_SHUTDOWN_TIMEOUT=60000
```

然后，停止并重新启动 processd (p. 44) 以使更改生效。

Windows 环境中的 SpectroSERVER 关闭

在 Windows 环境中，processd 会一直等到所有子进程均已关闭才停止。但是，Windows 注册表设置 WaitToKillServiceTimeout 设定了在启动 Windows 关闭之后 Windows 等待所有服务停止的时间长度。如果在经过 WaitToKillServiceTimeout 之后诸如 processd 之类的服务仍在运行，则 Windows 将终止此服务。默认值为 20 秒，该时间并不总是足以使 SpectroSERVER 在系统关闭时完全关闭。可以增加超时值。

遵循这些步骤:

1. 打开注册表编辑器。
2. 转到 HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control。
3. 单击 Control 项。
4. 在右窗格中双击 WaitToKillServiceTimeout 值。
5. 在窗口中，将值更改为不超过 600,000 毫秒（10 分钟）的任何值。
6. 单击“确定”。
7. 重新启动 Windows 服务器，以使更改生效。

设置格局映射条目超时

默认情况下，每个格局条目都无限期地保留在格局映射中。使用位置服务器配置文件 (.locrc)，可以指定一个时间，该时间之后格局条目将超时并从格局映射中自动删除。

注意：在以前的 CA Spectrum 版本中，格局条目在一小时后超时，并自动从格局映射中删除。从 CA Spectrum 9.2.2 开始，默认情况下格局条目不超时。

重要说明！我们建议*不要*使用超时值以从格局映射中自动删除条目。相反，请保留默认超时值。在需要时，使用 MapUpdate 命令从格局映射中删除条目。有关详细信息，请参阅*《数据库管理用户指南》*。

遵循这些步骤:

1. 导航到 <*$SPECROOT*>/LS 目录。
2. 在 .locrc 文件中更新或添加以下条目：

   MET_INTERVAL=*time_out_value*

   time_out_value

   指示格局条目将超时并从格局映射中自动删除之前经过的时间（以毫秒为单位）。

   默认值：0 毫秒（无超时）

从现有 DSS 设置进行格局映射的问题

症状：

我通过从其他安装复制来设置单独的分布式环境时，遇到格局映射问题。我注意到在主要 SpectroSERVER 关闭之后，无法切换到备用 SpectroSERVER，且出现用户安全性问题。

解决方案：

要复制格局映射，请删除对以前的 DSS 设置的所有引用。格局映射不得包含对以前的 DSS 设置的引用。使用以下过程可从现有的 DSS 设置映射格局。

遵循这些步骤:

1. 分离 SpectroSERVER，如下所示：
   a. 使每个 SpectroSERVER 成为独立服务器。

      位置服务器现在指向安装它的本地服务器。

   b. 验证每个 SpectroSERVER 的格局映射是否均不同。

      注意：在现有的 DSS 设置中，格局映射是包括所有服务器的单个映射。

2. 保存 SpectroSERVER 数据库。

   格局映射将是显而易见的。

3. 更改这些 SpectroSERVER 的 MLS 以还原原始的 MLS。

   注意：请勿更改原始的 DSS 设置。

4. 在不同的主机上重新加载每个 SpectroSERVER 的数据库。

5. 选择 SpectroSERVER 作为新的 DSS 设置中的 MLS，并使其他 SpectroSERVER 指向它。

   用户模型中的 dupModeList 将正确更新。

6. 删除格局映射中的备用条目（如果它们存在）。
7. 设置备用 SpectroSERVER 条目。
8. 在新的 DSS 中从主要 SpectroSERVER 加载数据库。

   将从现有的 DSS 映射格局。

详细信息：

设置分布式 SpectroSERVER 环境 (p. 29)

有关清除旧格局的问题

症状：

我们的生产 CA Spectrum 环境包括单个分布式安装。环境中包含多个 SpectroSERVER 和 OneClick 服务器，它们全都使用同一个主要位置服务器。

我们决定从生产环境中删除两个 SpectroSERVER 和一个 OneClick 服务器，以便建立一个独立的开发环境。我们对开发 SpectroSERVER 和 OneClick 服务器进行配置，让其参照新的位置服务器。我们更新了 .hostrc 文件，以便每个环境中的服务器仅可以访问各自的位置服务器。此外，我们在生产环境中更新了格局映射，以删除开发格局。

然而，当我查看开发 OneClick 服务器上“受监控的格局”列表时，仍能看到所有的旧生产格局。由于对 .hostrc 文件所做的更改，它们被列为“没有权限”。

解决方案：

根据您描述的设置，您现在有两个单独的环境。但是，您的开发格局映射缓存着陈旧的生产格局。您从生产环境中删除了陈旧的格局，但并未从开发环境中删除它们。因此，您必须手动从开发环境中删除陈旧的格局。

您可以选择两种方式删除陈旧的格局。

解决方案：

您可以使用以下命令从开发格局映射中删除陈旧的格局：

`MapUpdate -remove` *格局句柄*

解决方案：

您可以重新启动所有 SpectroSERVER 进程。有关详细信息，请参阅停止并重新启动 Processd (p. 44)。然后，重新启动 OneClick 服务器。有关 OneClick 服务器管理的详细信息，请参阅《管理员指南》。

在您重新启动服务器后，此问题便已得到解决。

第 3 章： 在分布式 SpectroSERVER 环境中跨防火墙进行通信

此部分包含以下主题：

跨防火墙进行通信

在许多网络环境中可以应用跨防火墙进行通信。在分布式环境中，防火墙可能会影响您的部署。

注意：在开始分布式 CA Spectrum 安装之前，配置防火墙以在端口 46517（端口 sradmin）上启用通信。

用于在 CA Spectrum 组件之间启用通信的选项取决于所部署的防火墙类型。其他因素（如成本和所需的安全级别）也会发挥作用。这些因素包括虚拟专用网络 (VPN)、节点到节点的通道或管道以及在数据包通过防火墙之前封装它们的代理。

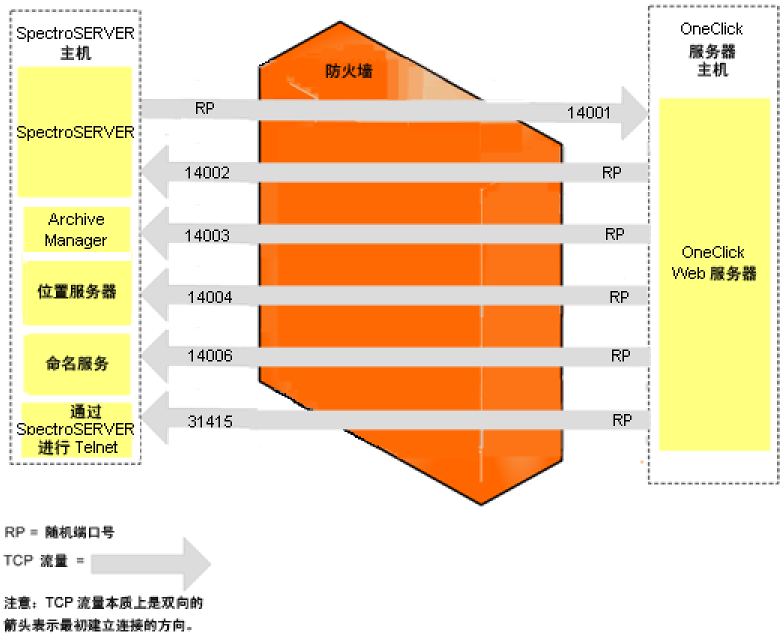
我们建议您与网络的防火墙管理员合作，制定出允许 CA Spectrum 和 OneClick 通信穿过防火墙的策略。

跨防火墙的 SpectroSERVER 和 OneClick Web 服务器通信

OneClick Web 服务器与 SpectroSERVER 主机系统上的进程进行通信来收集数据，以便显示在 OneClick 客户端中。OneClick Web 服务器通常启动此通信。

OneClick Web 服务器建立与特定 SpectroSERVER 主机端 TCP 端口的连接。Web 服务器使用这些端口发送请求和接收响应。但是，OneClick 使用单个侦听端口（默认为 14001）。SpectroSERVER 启动与该端口的连接。因此，通常需要修改防火墙配置。SpectroSERVER 使用双向 IIOP（Internet ORB 间协议）与其 CORBA 客户端进行通信。

下图说明了 OneClick Web 服务器与 SpectroSERVER 进行通信所需的 IP 连接。在使用 TCP 的所有情况下，从随机端口到特定的固定端口建立连接。



OneClick 默认端口和防火墙

HTTP 侦听端口

OneClick 用于 HTTP 通信的默认端口是端口 80。如果将 OneClick Web 服务器配置为使用端口 80 以外的端口，则防火墙也必须设置为允许此通信。

注意：有关详细信息，请参阅*《管理员指南》*和*《安装指南》*。

Windows XP SP2 平台上选择保持启用 Windows 防火墙的 OneClick 用户在运行 OneClick 控制台时可能会遇到问题。

注意：有关配置 Windows 防火墙的详细信息，请参阅 Microsoft 知识库文章 842242， 网址为 http://support.microsoft.com。

CORBA 侦听端口

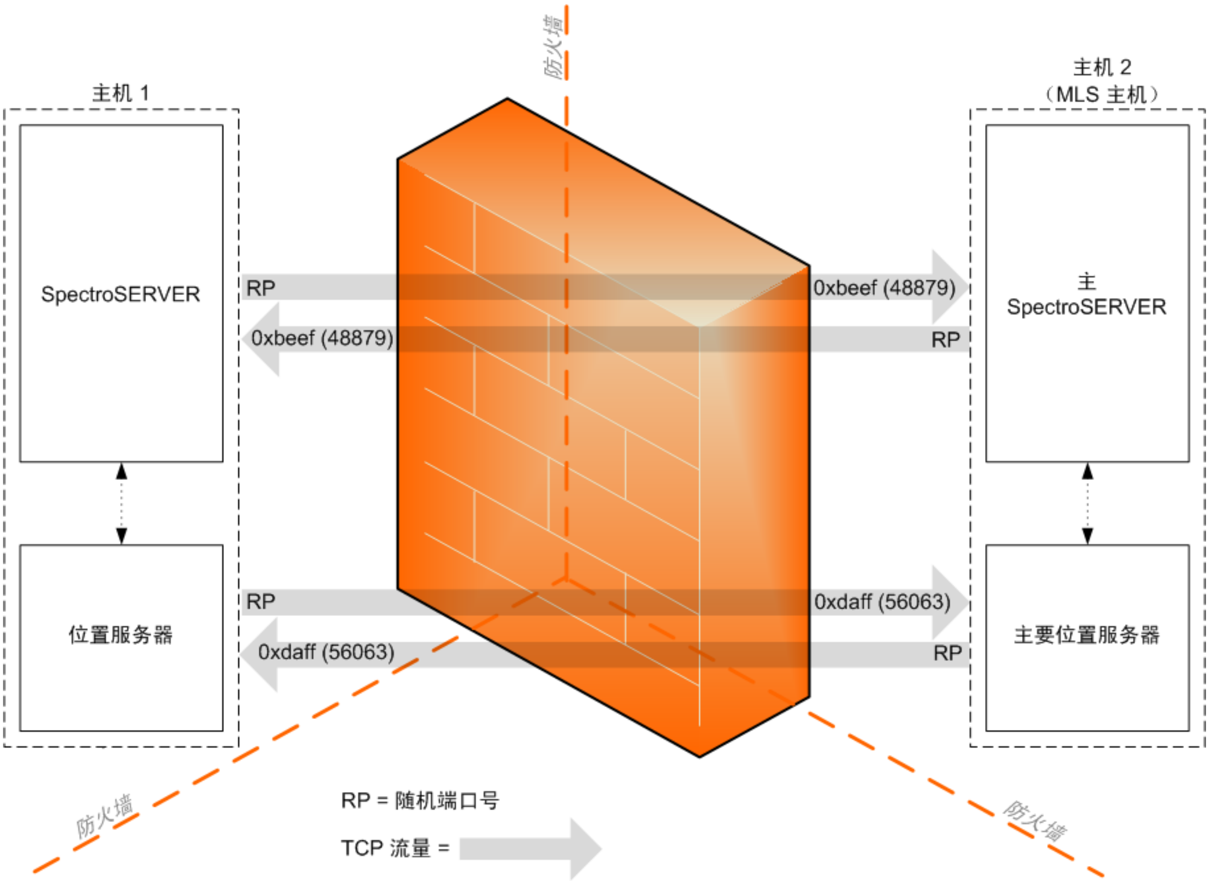
默认的 CORBA OneClick 服务器侦听端口值位于 <*$SPECROOT*>/tomcat/webapps/spectrum/META-INF/context.xml 中。

```
vbroker.se.iiop_tp.scm.iiop_tp.listener.port=<新端口号>
```

远程 SpectroSERVER 和防火墙

下图说明了两个远程 SpectroSERVER 通过防火墙进行通信时所需的 IP 连接。在使用 TCP 的所有情况下，从随机端口到特定的固定端口建立连接。

重要说明！不是主要位置服务器 (MLS) 的 SpectroSERVER 之间的所有通信都通过 MLS-SpectroSERVER 进行路由。因此，每个 SpectroSERVER 只能直接与 MLS-SpectroSERVER 进行通信。

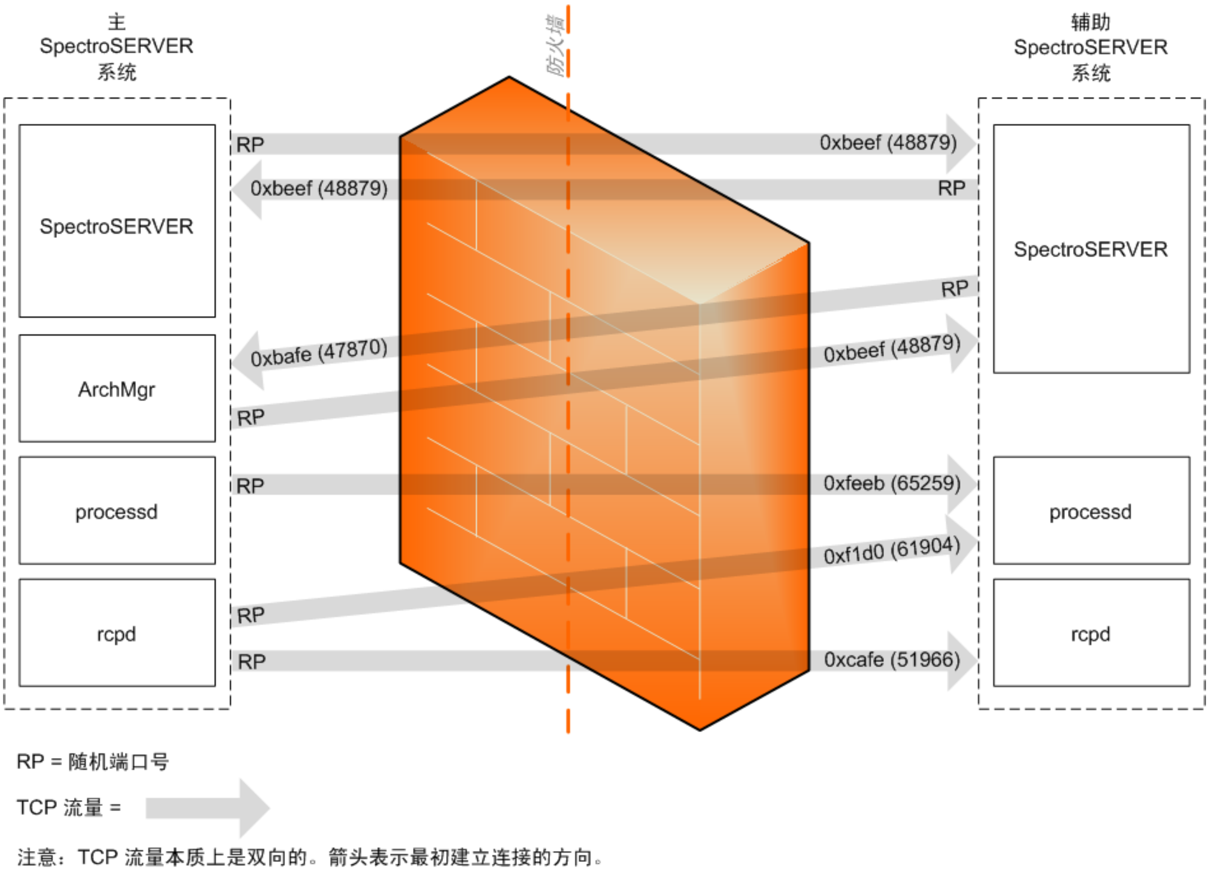


详细信息：

位置服务器 (p. 32)

跨防火墙的主要和备用 SpectroSERVER 通信

下图说明了容错环境中的主要和备用 SpectroSERVER 通过防火墙进行通信时所需的 IP 连接。在使用 TCP 的所有情况下，从随机端口到特定的固定端口建立连接。



适用于 NAT 防火墙环境的 CA Spectrum 配置文件

网络地址转换 (NAT) 用于构造具有一致的内部 IP 寻址和处理到 Internet 的 IP 地址转换的单个节点的网络。CA Spectrum 支持 NAT。可以在专用 IP 地址域中部署 CA Spectrum，并且可以在 NAT 防火墙之外与客户端保持连接。因此，DSS 环境现在可以包含由这些防火墙分隔的多个域。

NAT 环境的唯一要求是，客户端能够按名称解析服务器。在 NAT 的专用端上，主机名必须解析为专用端 IP 地址。在 NAT 的公共端上，主机名必须解析为公共端 IP 地址。

默认端口配置

可以更改 CA Spectrum 进程使用的默认端口或套接字号,以便进程在防火墙环境中正常工作。

可以更改以下进程的端口号或套接字号：

- OneClick WebServer (p. 55)
- SpectroSERVER (p. 57)
- Archive Manager (p. 57)
- 位置服务器 (p. 58)
- 命名服务 (p. 58)
- 远程复制进程后台进程 (rcpd) (p. 59)
- CLI 后台进程 (vnmshd) (p. 59)

注意： 您无法更改的唯一端口是，远程管理后台进程 sradmin 的端口和通过 SpectroSERVER 的 Telnet 的端口。

更改 SpectroSERVER 端口号

可以更改 SpectroSERVER 用于 CORBA 请求的端口号。

使用文本编辑器编辑 .vnmrc 文件以反映新的端口号。此文件位于 <*$SPECROOT*>/SS/ 中。输入以下命令：

```
orb_args=-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=<新端口号>
```

详细信息：

端口冲突解决方案 (p. 30)

更改 Archive Manager 端口号和套接字号

可以更改 Archive Manager 用于特定请求的端口号和套接字号。

使用文本编辑器编辑 .configrc 文件以反映新的端口号。此文件位于 <*$SPECROOT*>/SS/DDM 中。输入以下命令：

```
orb_args=-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=<新端口号>
```

要更改 Archive Manager 用于侦听来自 VNM 和 SSAPI 客户端的请求的套接字号，请使用文本编辑器编辑 .configrc 文件。此文件位于 <*$SPECROOT*>/SS/DDM 中。更改以下变量：

```
ARCH_MGR_SOCKET_NUMBER=<新端口号>
```

详细信息：

端口冲突解决方案 (p. 30)

更改位置服务器端口号和套接字号

可以更改位置服务器用于特定请求的端口号和套接字号。

要更改位置服务器用于 CORBA 请求的端口号，请使用文本编辑器编辑 .locrc 文件以反映新的端口号。此文件位于 <*$SPECROOT*>/LS 中。输入以下命令：

```
orb_args=-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=<新端口号>
```

要更改位置服务器用于 VNM 和 SSAPI 请求的套接字号，请使用文本编辑器编辑 .locrc 文件。更改以下变量：

```
LOC_SERVER_SOCKET_NUMBER=<新端口号>
```

详细信息：

端口冲突解决方案 (p. 30)

更改 Visibroker 命名服务端口号

可以更改 Visibroker 命名服务使用的端口号。

注意：默认端口号是 14006。

遵循这些步骤:

1. 右键单击“我的电脑”，然后选择“属性”。

   将打开“系统属性”对话框。

2. 单击“高级”选项卡，然后单击“环境变量”按钮。
3. 选择 NAMING_SERVICE_PORT，然后编辑它以反映新的端口号：

   ```
   NAMING_SERVICE_PORT=<新端口号>
   ```

要在 Solaris 上更改 Visibroker 命名服务端口号，请在 spectrum60.env 文件中设置环境变量。此文件位于 /opt/SPECTRUM 目录中。更改以下变量：

NAMING_SERVICE_PORT=<*新端口号*>

远程复制进程后台进程 (rcpd) 端口号配置

远程复制进程后台进程 (rcpd) 端口号可通过 rcpd_comm_port .vnmrc 资源进行配置。

CLI 后台进程 (vnmshd) 端口号配置

CLI 后台进程 (vnmshd) 端口号可通过 vsh_tcp_port 参数进行配置。

注意：有关 vsh_tcp_port 参数的信息，请参阅*命令行界面用户指南*。

详细信息：

端口冲突解决方案 (p. 30)

第 4 章： 容错

此部分包含以下主题：

关于容错

容错需要多个 SpectroSERVER 来管理给定的格局。在每个 SpectroSERVER 上加载该格局的数据库副本。但是，在任何时间只有一个副本处于活动状态。具有活动数据库的 SpectroSERVER 称为*主要 SpectroSERVER*。非活动数据库在待机 SpectroSERVER（即备用 SpectroSERVER）上运行。也可以在第三极 SpectroSERVER 上安装数据库的其他非活动副本。

如果主要 SpectroSERVER 失败，则备用 SpectroSERVER 上的数据库将变为活动状态，并且备用 SpectroSERVER 开始管理网络。连接到主要 SpectroSERVER 的应用程序将自动切换到备用 SpectroSERVER。在主要 SpectroSERVER 恢复使用时，应用程序将自动切换回主要 SpectroSERVER，备用 SpectroSERVER 再次变为非活动状态。

注意： 从备用 SpectroSERVER 运行应用程序时，并非所有的应用程序都可以发挥其全部功能。设置容错环境的主要原因是确保网络的持续监控，而不是创建 CA Spectrum 的完整副本。

详细信息：

容错设置中的陷阱定向器 (p. 79)

容错环境中的 SpectroSERVER 优先级

管理同一格局的主要、备用和第三级 SpectroSERVER 必须都具有相同的格局句柄和相同的建模目录。这些服务器通过优先级数值来相互区分。最小数指示主要 SpectroSERVER。SpectroSERVER 在安装时的默认优先级值为 10。要将 SpectroSERVER 指定为备用服务器，请为其分配较大的优先级数，如 20。同样，第三级 SpectroSERVER 应该具有比备用服务器更大的优先级数，例如 30。

首次设置容错环境时，可以在任何待机 SpectroSERVER 上使用 SSdbload 实用工具 (p. 68)加载数据库副本时分配优先级值。

要稍后更改优先级值，可以使用“已加载格局”子视图。通过在“导航”面板中选择本地格局，然后在“组件详细信息”面板中选择“信息”选项卡，可以访问此子视图。

注意：“已加载格局”子视图与“SpectroSERVER 控制”子视图不同。通过在“导航”面板中选择“VNM”，然后在“组件详细信息”面板中选择“信息”选项卡，可以访问“SpectroSERVER 控制”子视图。

详细信息：

建立容错 (p. 68)

数据同步

在容错 CA Spectrum 环境中，单个数据库在任何给定时间都处于活动状态。因此，必须定期更新其他数据库，以反映新模型和对活动数据库中属性值的更改。此数据同步是通过 CA Spectrum 联机备份功能完成的。可以根据需要或者按有规律排定的间隔来运行联机备份。对主要 SpectroSERVER 运行联机备份时，它将创建当前数据库的备份副本。联机备份会将副本自动加载到每个指定的备用 SpectroSERVER 上。

与在任何 DSS 环境中一样，容错环境中的每个 SpectroSERVER 都必须安装有相同的建模目录。联机备份将复制当前的建模目录。但是，它不复制所有的 .i 文件或者与单个管理模块关联的其他元素。因此，如果在主要 SpectroSERVER 上安装任何新管理模块，则还要在任何备用 SpectroSERVER 上安装相同的新管理模块。

注意：有关详细信息，请参阅*《数据库管理指南》*。

备用 SpectroSERVER 轮询主要 SpectroSERVER 以获取状态信息时，在 <*$SPECROOT*>/custom/Events 目录中定义的 EventDisp 和 Alertmap 文件将传播到容错服务器。

如果备用 SpectroSERVER 未重新启动，则生成警报

主要 SpectroSERVER 将其数据库与备用 SpectroSERVER 同步时，将生成“与备用服务器失去联系 (0x00010c0e)”事件和警报。备用 SpectroSERVER 已被关闭以便从主要 SpectroSERVER 加载新的数据库。

可以设置处理此警报的规则，以便仅当备用 SpectroSERVER 未重新启动时才生成警报。

通过 EventPair 规则您可以指定：如果发生“与备用服务器失去联系”事件且在指定的时间段内未发生“与备用服务器建立联系 (0x00010c0f)”事件，则生成新事件。然后，可以指定此新事件创建一个事件和警报，指示备用 SpectroSERVER 仍处于关闭状态。

遵循这些步骤:

1. 用文本编辑器打开 EventDisp 文件。

   注意：EventDisp 文件位于 *<$SPECROOT>*/SS/CsVendor/Cabletron 目录中。

2. 查找内容为 0x00010c0e E 50 A 2, 0x00010c0e 的行，并将此行更改为以下内容：

   ```
   0x00010c0e R Aprisma.EventPair, 0x00010c0f,
       <numberofsecondstowait><generatedeventcode>
   ```

   <generatedeventcode>

   是备用 SpectroSERVER 未在 *<numberofsecondstowait>* 中指定的时间内启动时生成的事件代码。

3. 将以下行添加到 EventDisp 文件中：

   ```
   <generatedeventcode>E 50 A 2, <generatedalarmcode>
   ```

   <generatedeventcode>

   是备用 SpectroSERVER 未启动时在第 2 步中生成的事件代码。“E 50”指示事件已记录，其重要级别值为 50。A 2 指示创建了主要警报。*<generatedalarmcode>* 是基于此事件生成的警报代码。

4. 为此警报创建“可能原因”文件，指示在数据同步之后，尚未重新建立与备用 SpectroSERVER 的联系。

注意： 有关详细信息，请参阅《*事件配置用户指南*》。

备用 SpectroSERVER 就绪级别

备用 SpectroSERVER 被视为处于三种不同的就绪级别之一。就绪状态取决于服务器配置和状态。就绪级别的定义如下：

热

备用 SpectroSERVER 正在运行，并可用于在主要 SpectroSERVER 发生故障时立即接管，因为它已经在轮询。要配置备用 SpectroSERVER 以达到该就绪级别，请将以下行添加到 .vnmrc 文件：secondary_polling=yes。此语句可导致待机 SpectroSERVER 每次启动时都开始轮询和处理陷阱，而不管其与主要 SpectroSERVER 的连接状态如何。

温

备用 SpectroSERVER 正在运行，但是该服务器可以在很短的时间内变为完全可用。备用 SpectroSERVER 尚未配置为在失去与主要 SpectroSERVER 的联系*之前*开始轮询。例如，它在 .vnmrc 文件中没有 secondary_polling 条目，或者该条目设置为 no。

如果 secondary_polling 条目不在 .vnmrc 文件中或者该条目设置为 no，则备用 SpectroSERVER 在处于待机模式时不处理陷阱。

冷

备用 SpectroSERVER 未运行，必须在主要 SpectroSERVER 发生故障时启动它。在这种情况下，是否为备用轮询配置了备用 SpectroSERVER 是不相关的。

SpectroSERVER 警报同步

主要和备用 SpectroSERVER 使用全局警报服务 (GAS) 连接来共享警报信息。这些 SpectroSERVER 使用警报信息同步警报。此同步有助于防止重复的警报。

容错警报同步的选项包括启用服务和调试日志记录。.vnmrc 文件中的设置控制这些选项。有关详细信息，请参阅容错警报服务 (.vnmrc) 资源 (p. 21)。

部署 VHM 和机箱设备时，如果配置有备用 SpectroSERVER，警报同步可能会出现一些意外的行为。当主要 SpectroSERVER 宕机时，在主要 SpectroSERVER 上关联的警报不会在备用 SpectroSERVER 上关联。而且，您会看到针对每种不同状况或症状的警报。当主要 SpectroSERVER 重新联机时，会相应地发生警报关联。但是，仍不会在机箱设备上执行关联。

不保留关联是因为备用 SpectroSERVER 上的警报是生成的，而不是根据事件创建的。

根据同步是从主要 SpectroSERVER 到备用 SpectroSERVER 还是从备用 SpectroSERVER 到主要 SpectroSERVER，警报同步的过程有所不同。以下部分介绍了其中每个方案：

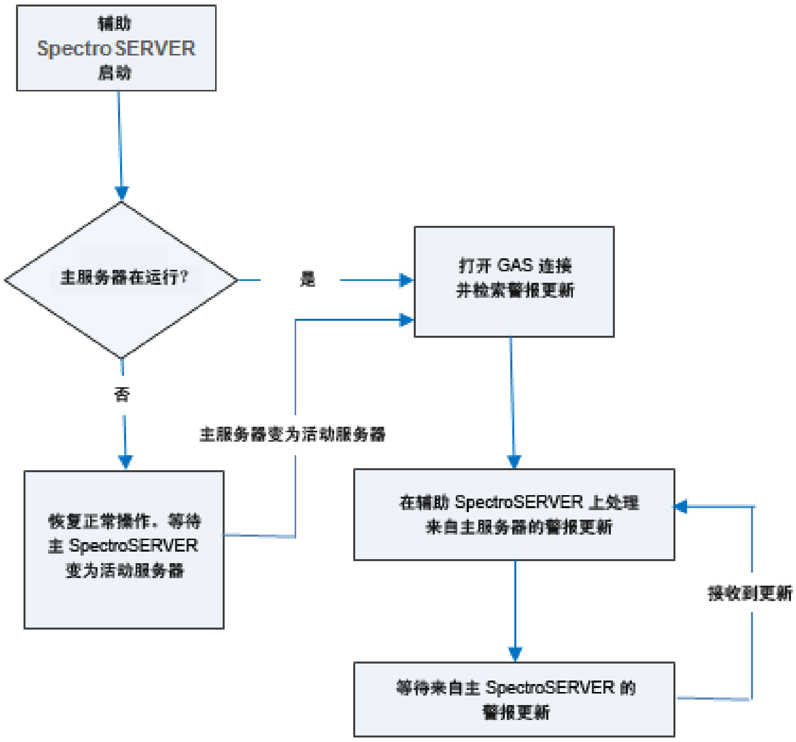
- 从主要 SpectroSERVER 到备用 SpectroSERVER 的同步 (p. 64)
- 从备用 SpectroSERVER 到主要 SpectroSERVER 的同步 (p. 67)

从主要 SpectroSERVER 到备用 SpectroSERVER 的同步

如果 SpectroSERVER 作为备用 SpectroSERVER 运行，则它尝试打开与主要 SpectroSERVER 的 GAS 连接。如果连接成功，则备用 SpectroSERVER 将进行注册以便从主要 SpectroSERVER 接收警报更新。备用服务器与主要服务器保持连接，并在主要服务器上出现警报更新时继续接收它们。否则，备用 SpectroSERVER 尝试每分钟连接一次，直到主要 SpectroSERVER 变为活动状态为止。

注意：在主要服务器和备用服务器上都必须启用容错警报服务，备用 SpectroSERVER 才能尝试连接到主要 SpectroSERVER 以进行警报同步。要控制此功能，请使用 .vnmrc 文件中的 ftasv_enabled 参数。有关详细信息，请参阅容错警报服务 (.vnmrc) 资源 (p. 21)。

下图说明了备用 SpectroSERVER 如何处理来自主要 SpectroSERVER 的警报更新：



- 在备用 SpectroSERVER 上添加新的主要 SpectroSERVER 警报，并将其标记为“过时”。如果新警报与现有警报等效，则前者将替换后者。通过验证以下参数，可以确定警报是否等效：
  - 唯一警报 (ID)
  - 模型句柄
  - 可能原因
  - 警报标识符

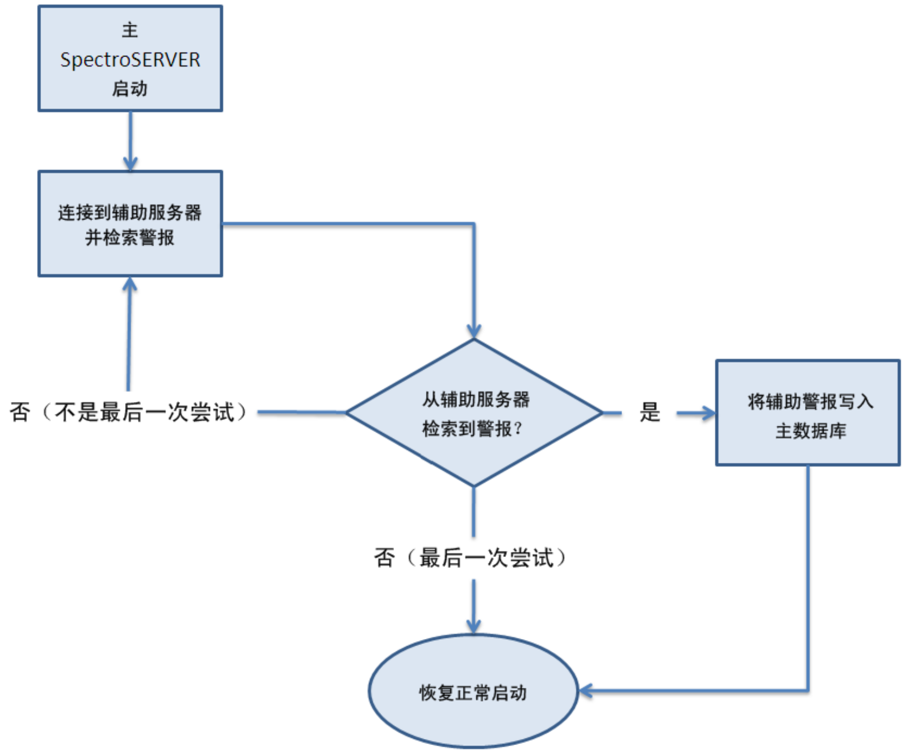
  如果两个警报具有相同 ID，则认为它们是等效的。除非警报具有不同的警报标识符，否则处于同一模型上（即，它们具有相同的模型句柄）且具有相同的可能原因的警报也视为等效。
- 来自主要 SpectroSERVER 的警报属性更新将应用于备用 SpectroSERVER 上的相同警报。如果在主要 SpectroSERVER 上清除了某个警报，则在备用 SpectroSERVER 上也将清除它。
- 对于维护模式警报同步，会更新 isManaged 和 isNotHibernating 属性。

从备用 SpectroSERVER 到主要 SpectroSERVER 的同步

主要 SpectroSERVER 在故障或计划内维护之后联机时，它将连接到备用 SpectroSERVER，并获取所有的可用警报。如果第一次连接尝试失败，则主要服务器可以进行多次连接尝试以便同步。主要 SpectroSERVER 的 .vnmrc 文件中的 ftasv_max_conn_retry_count 和 ftasv_conn_retry_interval 参数控制尝试次数和频率。有关详细信息，请参阅容错警报服务 (.vnmrc) 资源 (p. 21)。

重要说明！如果在主要服务器尝试连接到备用 SpectroSERVER 时后者未运行，则主要服务器会耗尽其同步尝试。结果是启动主要服务器出现延迟。此延迟是不可避免的，因为它发生在模型激活之前。通过验证主要服务器启动时备用服务器是否正在运行，可以避免该情况。也可以减少重试计数或间隔大小以减少潜在的延迟。或者，可以使用 .vnmrc 文件中的 ftasv_enabled 参数禁用容错警报服务。

下图说明了主要 SpectroSERVER 如何打开到备用 SpectroSERVER 的 GAS 连接并检索其所有警报。



在成功连接后：

- 在检索到警报列表之后，主要 SpectroSERVER 将警报写入其数据库。主要 SpectroSERVER 继续启动，并且从数据库读取这些警报。
- 在检索时，来自备用 SpectroSERVER 的警报将替换存储在主要数据库中的警报。

- 在正确的模型上处理和断言从数据库读取的所有警报。
- 在主要 SpectroSERVER 关闭时，为在备用 SpectroSERVER 上生成的每个新警报生成备用“警报设置”事件 (0x10714)。
- 在主要 SpectroSERVER 关闭时，为在备用 SpectroSERVER 上清除的每个警报生成备用警报清除事件 (0x10715)。

注意：对于蓝色和灰色警报，不会发生同步。对于褐色警报，会发生同步，但 WA_Link 模型除外。WA_Link 模型例外仅适用于褐色警报。

在失败的同步之后：

- 如果在主要 SpectroSERVER 尝试连接到备用 SpectroSERVER 时警报同步失败，则主要 SpectroSERVER 将启动，但是在主要 LocalLscpe 模型上生成事件和关键警报 (0x10c35)。事件和警报包括每次尝试失败的原因。要在警报同步失败之后继续运行主要 SpectroSERVER，请运行联机备份以使主要和备用 SpectroSERVER 数据库同步。

建立容错

在创建任何模型之前，首次安装 CA Spectrum 时可以设置容错环境。或者，可以在安装 CA Spectrum 之后设置容错环境。

以下过程介绍了如何设置两个 SpectroSERVER：主要和备用。也可以通过采取相同的步骤来设置第三级 SpectroSERVER。但是，分配给第三级 SpectroSERVER 的优先级数要比备用 SpectroSERVER 的大。

注意： 要在具有 Southbound 网关集成的环境中建立容错，请参阅《*Southbound Gateway Toolkit 指南*》。

遵循这些步骤:

1. 在主要 SpectroSERVER 和备用 SpectroSERVER 上安装具有相同建模目录的相同版本的 CA Spectrum。每个服务器都需要相同的格局句柄。
2. 验证主要和备用 SpectroSERVER 在其 .hostrc 文件中是否都有提供 SpectroSERVER 相互访问权限的条目。

   注意：如果在主要 SpectroSERVER 上的 .hostrc 文件中为备用 SpectroSERVER 指定安全用户，并且备用 SpectroSERVER 正在 Windows 环境中运行，则在安全用户列表中包括用户 SYSTEM。

3. 验证备用 SpectroSERVER 服务器上 .locrc 文件中的 MAIN_LOCATION_HOST_NAME 参数是否指向与主要 SpectroSERVER 上的 .locrc 文件相同的系统名称。否则，同步将失败。

4. 配置主要和备用 SpectroSERVER 以便运行每个 SpectroSERVER 的用户是相同的。如果用户不同，则备用 SpectroSERVER 将失败或者在联机备份之后无法正常运行。
5. 通过运行联机备份创建主要 SpectroSERVER 数据库的副本。或者，如果 SpectroSERVER 已关闭，则使用带 -cm 参数的 SSdbsave 实用工具（以保存建模目录和任何新模型）。

   注意：有关详细信息，请参阅*《数据库管理指南》*。
6. 验证所创建的保存文件是否可用于承载备用 SpectroSERVER 的服务器。如有必要，请将该文件复制到服务器。
7. 在备用服务器上，在 SpectroSERVER 已关闭的情况下，导航到 CA Spectrum SS 目录，并使用以下命令加载保存文件：

   ```
   ../SS-Tools/SSdbload -il -add precedence savefile
   ```

   precedence

   指定一个大于主要服务器默认值 10 的数值（建议指定 20）。

   savefile

   指定以前创建的保存文件的名称。
8. （可选）将“secondary_polling=yes”一行添加到 .vnmrc 文件，以允许备用 SpectroSERVER 充当“热”备份 (p. 64)。
9. 启动主要 SpectroSERVER（如果它尚未运行）。
10. 启动备用 SpectroSERVER。
11. 要验证设置，请使用带 view 参数的 MapUpdate 命令以显示当前的格局映射。

    注意：有关详细信息，请参阅*《数据库管理指南》*。

    如果主要 SpectroSERVER 失败，则备用 SpectroSERVER 现在可用于自动接管。如果以前激活了备用轮询，则备用 SpectroSERVER 立即可用。否则，服务器在检测到它已与主要 SpectroSERVER 失去联系后，会立即开始轮询。

服务从主要 SpectroSERVER 切换到备用 SpectroSERVER 时，“连接状态”图标  将呈黄色。要查看格局中所有服务器的连接状态，请单击“连接状态”图标。在“连接状态”对话框中，格局中每个服务器的“连接状态”图标都呈黄色，以指示“已切换”状态。

主要 SpectroSERVER 重新联机时，备用 SpectroSERVER 停止轮询（除非已将 secondary_polling 设置为“yes”）。所有的应用程序都将切换回主要 SpectroSERVER。但是，在备用 SpectroSERVER 处于活动状态时对其进行的任何编辑都*不*会自动复制到主要 SpectroSERVER。在主要 SpectroSERVER 上手动重新创建这些修改。

在重新启动主要 SpectroSERVER 时，加载所有模型后会立即接受连接，但*前提*是已激活所有模型。激活模型可能需要一些时间。由于重新启动主要 SpectroSERVER 后备用 SpectroSERVER 会立即停止轮询，因此可能会导致网络管理覆盖面出现空隙。

为避免此情况，请在主要 SpectroSERVER 上编辑 .vnmrc 文件，以便将 wait_active 资源设置为“yes”。此参数可导致服务器在接受任何连接之前一直等待，直到激活所有模型。CA Spectrum 控制面板中的消息区域还动态显示已激活模型的百分比。SpectroSERVER 看起来可能需要更长的时间来启动。但是，在所有模型都已激活时，SpectroSERVER 就可以管理网络了。

在备用 SpectroSERVER 上也可以将 wait_active 资源设置为“yes”。在主要 SpectroSERVER 的计划内关闭期间，这样就可以在 CA Spectrum 控制面板中验证备用 SpectroSERVER 是否准备接管。

注意：有关详细信息，请参阅*《数据库管理指南》*。

验证容错配置

在分布式 SpectroSERVER 部署中设置容错之后，验证 OneClick 服务器是否有权访问主要和备用 SpectroSERVER。如果没有连接到这两个服务器，则 OneClick 服务器无法故障切换到备用 SpectroSERVER。

遵循这些步骤:

1. 访问“OneClick 管理”的“格局”网页。
2. 检查“备用状态”列。验证 OneClick 是否已与备用 SpectroSERVER 建立联系。

   状态还指示容错是否准备进行故障转移。

   容错配置将得到验证。

测试容错

在初始安装期间，备用 SpectroSERVER 可能无权访问主要 SpectroSERVER 有权访问的所有设备。此情况可导致备用 SpectroSERVER 生成假警报。为避免假警报，请通过测试容错来验证备用 SpectroSERVER 是否可以管理网络设备。

注意： 每当将新设备添加到主要 SpectroSERVER 时，都会测试容错。

遵循这些步骤:

1. 在主要和备用 SpectroSERVER 都已启动并运行时，关闭主要 SpectroSERVER。

   “连接状态”图标  呈黄色，以指示“已切换”状态。

   红色的连接器指示，OneClick 服务器无法联系备用 SpectroSERVER。

2. 等待 15 - 20 分钟，以便备用 SpectroSERVER 运行。

3. 验证以下情况：

   - “连接状态”图标未呈红色。
   - 所有的设备模型和可 ping 的模型都保持 SNMP 或 ICMP 联系。

     如果失去此联系，请验证备用 SpectroSERVER 是否有权访问您的设备。如果适用，请联系网络管理员以解决此问题。

   - CA Spectrum 正在管理具有“已建立联系”状态的所有设备。通过从任何设备模型检查设备联系或管理联系丢失警报，对状态进行验证。

4. 重新启动主要 SpectroSERVER。

   “连接状态”图标呈绿色，以指示正常联系状态。

重新启动主要 Archive Manager 和主要 SpectroSERVER

为防止在容错 SpectroSERVER 环境中丢失任何事件或统计数据，请让单个 Archive Manager 运行。在主要 SpectroSERVER 主机上安装此主要 Archive Manager。通过此配置，在发生故障时，不会丢失数据。

可能有以下两种故障情况：

- 主要 SpectroSERVER 停止。然后，备用 SpectroSERVER 将事件和统计信息转发到在承载主要 SpectroSERVER 的服务器上运行的主要 Archive Manager。在主要 SpectroSERVER 重新启动时，未丢失事件和统计数据。
- 运行主要 SpectroSERVER 和主要 Archive Manager 的计算机将完全停止运行。然后，备用 SpectroSERVER 在其数据库中缓存事件和统计数据，直到主要 SpectroSERVER 计算机重新联机。

如果其服务器已关闭，或者如果主要 SpectroSERVER 停止运行，请重新启动主要 Archive Manager 和主要 SpectroSERVER。

遵循这些步骤:

1. 在主要 SpectroSERVER 上启动 CA Spectrum 控制面板，并从“控制”菜单中选择“启动 Archive Manager”。

   在主要 Archive Manager 重新运行时，备用 SpectroSERVER 会将其缓存的事件和统计数据转发到主要 Archive Manager。

   重要说明！为避免数据丢失，在备用 SpectroSERVER 已转发缓存的数据之前，请勿启动主要 SpectroSERVER。

2. 在备用 SpectroSERVER 上访问“事件和统计日志信息”视图：

   a. 突出显示 VNM 图标。

   b. 从“图标子视图”菜单中选择“配置”。

   c. 在“格局配置”视图中单击“事件/统计日志”。

3. 监控“事件和统计日志信息”视图。

   在“本地存储的事件”和“本地存储的记录”字段都显示零时，在备用 SpectroSERVER 的数据库中缓存的事件和统计数据都已转发到主要 Archive Manager。

4. 重新启动主要 SpectroSERVER。

更改主要和备用 SpectroSERVER 的主机名

容错环境中的 SpectroSERVER 使用与其主机名关联的优先级值来识别彼此的关联关系。因此，要保持容错关联关系，请使用 SSdbsave 和 SSdbload 更改主要 SpectroSERVER 的主机名。

遵循这些步骤:

1. 使用带 -cm 选项的 SSdbsave 保存数据库。

2. 更改主机名。

3. 使用在第一步中创建的保存文件重新加载数据库。运行带 -il 选项和 -replace 选项的 SSdbload：

   ```
   SSdbload -il -replace precedence savefile
   ```

   此命令可导致数据库将新的主机名与指定主要 SpectroSERVER 的优先级值 (10) 关联。

   下次因运行联机备份而同步数据库时，主机名的更改将传达到所有温待机或热待机 SpectroSERVER。

   但是，在此期间，主机名更改可防止待机 SpectroSERVER 检测到主要 SpectroSERVER 正在运行。因此，配置为温待机的任何 SpectroSERVER 都将开始轮询。

4. 使用带 -il 和 -replace 选项的 SSdbload 在温待机 SpectroSERVER 上加载保存文件，并指定一个可将它指定为待机 SpectroSERVER 的较大优先级值（例如 20）。

现在，可以更改备用 SpectroSERVER 的主机名。

遵循这些步骤:

1. 使用带 -cm 选项的 SSdbsave 保存数据库。
2. 对主机名进行更改。
3. 使用在第一步中创建的保存文件重新加载数据库。运行带 -il 选项和 -replace 选项的 SSdbload：

   ```
   SSdbload -il -replace precedence savefile
   ```

   此命令可导致数据库将新的主机名与指定备用 SpectroSERVER 的优先级值 (20) 关联。

   在重新启动备用 SpectroSERVER 时，服务器将新主机名和优先级传达到主要 SpectroSERVER。

注意：有关详细信息，请参阅 *《数据库管理指南》*。

监控主要和备用 SpectroSERVER 之间的转换

可以使用监视来监控容错环境的状态。创建一个监视，以便在主要或备用 SpectroSERVER 准备接管向您报警。

遵循这些步骤:

1. 在 VNM 模型上创建监视以监控 PercentInitialized (0x11da6) 属性。

   在此属性的值等于 100% 时，SpectroSERVER 已初始化并准备接管。

2. 设置监视以在以下表达式的计算结果为 TRUE 时生成事件或警报或者运行脚本：

   ```
   PercentInitialized == 100
   ```

3. 将监视设置为活动的已轮询监视。
4. 使备用 SpectroSERVER 与主要 SpectroSERVER 同步以传播监视。
5. 在监视表达式中指定 Model_Name (0x1006e) 属性的值。仅当备用 SpectroSERVER 准备接管时，此属性才通知您。

   例如，如果以下监视表达式的计算结果为 TRUE，则备用 SpectroSERVER <*sec_server*> 准备接管。

   ```
   (PercentInitalized == 100) & (Model_Name= <sec_server>)
   ```

   注意：有关详细信息，请参阅*《监视用户指南》*。

6. 将以下行添加到备用 SpectroSERVER 上的 .vnmrc 文件，以限制发生假事件或警报的可能性：

   ```
   is_secondary = TRUE
   ```

   除非 CA Spectrum 确定备用 SpectroSERVER 已作为主要 SpectroSERVER 接管，否则此设置允许备用 SpectroSERVER 丢弃事件。

如何监控备用 SpectroSERVER 状态

可以创建一个监视，以便在备用 SpectroSERVER 服务器充当主要 SpectroSERVER 时向您报警。

遵循这些步骤:

1. 在 VNM 模型上创建一个监视，以监控备用 SpectroSERVER 的 PausePolling 属性 (0x11b63) 的值。

   此属性在备用 SpectroSERVER 上设置为 FALSE 时，备用 SpectroSERVER 轮询并充当主要 SpectroSERVER。

   例如，如果以下监视表达式的计算结果为 TRUE，则备用 SpectroSERVER <*sec_server*> 充当主要 SpectroSERVER。

   ```
   !PausePolling & (Model_Name == <sec_server>)
   ```

2. 将监视设置为活动的已轮询监视。
3. 使备用 SpectroSERVER 与主要 SpectroSERVER 同步以传播监视。

   注意：有关详细信息，请参阅*《监视用户指南》*。

4. 将以下行添加到备用 SpectroSERVER 上的 .vnmrc 文件，以限制发生假事件或警报的可能性：

```
is_secondary = TRUE
```

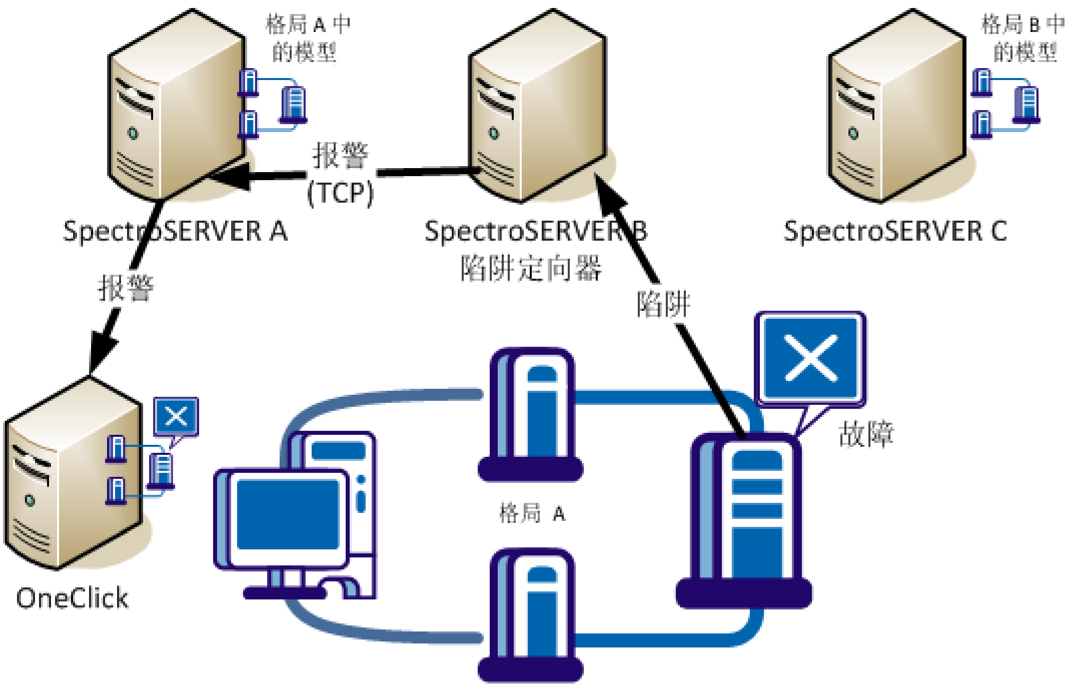
除非 CA Spectrum 确定备用 SpectroSERVER 已作为主要 SpectroSERVER 接管，否则此设置允许备用 SpectroSERVER 丢弃事件。

第 5 章： 处理陷阱定向器

此部分包含以下主题：

陷阱定向器

陷阱定向器是可以在分布式 SpectroSERVER 环境中启用的 SpectroSERVER 功能。陷阱定向器在陷阱定向器主机服务器上接收陷阱并处理它们。然后，陷阱定向器使用陷阱中的信息创建报警，该报警将通过 TCP 连接转发到 SpectroSERVER。因此，主要的陷阱定向器机制实际上是 CA Spectrum 内部报警，SpectroSERVER 将其发送给远程主机上的模型以便进一步处理。下图说明了陷阱定向器体系结构：



要使用陷阱定向器，请将一个 SpectroSERVER 指定为陷阱定向器服务器。在该服务器上启用陷阱定向器 (p. 80)。然后，将该服务器配置为从远程格局中建模的设备接收陷阱的网络管理站 (NMS)。

陷阱定向器支持加密的 SNMPv3 陷阱。CA Spectrum 尝试使用 SNMPv3 配置文件对传入的加密 SNMPv3 陷阱解密。

可用的 SpectroSERVER SNMPv3 配置文件被视为未知 SNMPv3 陷阱的凭据模板。收到加密的 SNMPv3 陷阱时，会使用 SNMPv3 配置文件对陷阱解密。

注意： 如果陷阱定向器格局具有包含有效凭据的 SNMPv3 配置文件，则作为陷阱转发器运行的 SpectroSERVER 可以处理 SNMPv3 陷阱。

陷阱和内存使用

启用陷阱定向器时，可以监控 SpectroSERVER 进程的内存使用。如果有大量陷阱（每秒超过 100 个）被转发到多个远程格局，便可能会发生 SpectroSERVER 无法转发所有传入陷阱的情况。然后，这些陷阱被排入队列以等待进一步处理。如果此情况持续存在，队列使用的内存可能会耗尽供 SpectroSERVER 进程使用的内存资源。随后，SpectroSERVER 就有可能会因为缺少足够内存而意外宕机。

除进程内存使用之外，还应密切监控 VNM 模型中的以下属性：

`alert_remote_fwd_queue_length`（属性 ID：0x130c3）

通常，如果队列大小不断增长而超过 1000 个元素，我们不建议您在环境中启用陷阱定向器。

陷阱数据通信合并

在网络增长提示您在其他格局之间分配模型时，可以保留原始格局主机作为这些模型的陷阱目标。这样，就可以避免在关联的设备上重新配置新陷阱目标。相反，可以合并多个陷阱目标，并且可以指定单个 SpectroSERVER 来接收和路由陷阱。如果负载共享是重要因素，则可以在多个服务器上启用陷阱定向器。每个服务器都处理来自一组设备的陷阱，这些设备配置为将陷阱发送到该服务器。

陷阱定向器维护最新的模型地址缓存。通过此缓存，陷阱定向器可以将陷阱源与模型匹配。陷阱定向器使用匹配操作将陷阱转发到远程格局上的目标模型。通过使缓存信息保持最新，陷阱定向器确保 CA Spectrum 可以为模型生成事件，而不管其位置如何。

注意： 向分布式环境添加新格局时，会影响陷阱定向器性能。为许多新模型处理大量的陷阱时，也会影响性能。在这些情况下，陷阱通知延迟时间可能会增加。

陷阱定向器如何更新地址缓存

陷阱定向器在分布式环境中维护模型的 IP 地址和位置的缓存。地址缓存充当索引，用于确定将报警转发到的位置。可以将模型定期添加到格局，将其从格局中删除，以及在格局之间移动它们。因此，陷阱定向器也定期更新缓存以使其内容保持最新。满足您可以指定的保留期限（或老化）阈值的记录将被删除。陷阱定向器还执行跨格局搜索，以查找缓存中不可用的模型 IP 地址（在它从这些 IP 地址接收陷阱时）。

以下步骤介绍陷阱定向器如何确定陷阱的目标模型：

1. 陷阱定向器服务器接收陷阱。
2. 陷阱定向器将陷阱中包括的 IP 地址与缓存记录中的 IP 地址进行比较。
3. 如果陷阱定向器找到匹配项，就会把报警转发到远程格局上的模型。

   否则，陷阱定向器执行以下任务，以确定目标模型并转发报警：

   - 陷阱定向器在已知格局中搜索匹配的 IP 地址。
   - 在它找到与远程格局上的 IP 地址关联的目标模型时，陷阱定向器用模型信息更新缓存。随后，它将报警转发到该模型。
4. 如果因跨格局搜索未找到匹配地址而将陷阱丢弃，则 CA Spectrum 在陷阱定向器服务器的 VNM 模型上生成事件。该事件指示未找到目标模型。

   例如，如果模型已删除，则找不到匹配项。

容错设置中的陷阱定向器

如果要实施容错陷阱定向器设置，则必须将设备配置为同时向主要和备用 SpectroSERVER 转发陷阱。备用服务器只有在检测到主要服务器已失败时才路由陷阱。在主要备份期间或同步过程中，备用 SpectroSERVER 接收陷阱定向器设置。

详细信息：

关于容错 (p. 61)

陷阱风暴设置

为了处理陷阱风暴，陷阱定向器使用在已建模设备上配置的陷阱风暴设置。陷阱定向器在检测到陷阱风暴时，会停止将报警转发到远程格局上的模型。陷阱定向器还断言模型的陷阱风暴警报。

陷阱风暴可能源自未在 CA Spectrum 中建模的设备。对于这些风暴，陷阱定向器使用在陷阱定向器服务器上为 VNM 模型配置的陷阱风暴处理设置。

启用和禁用陷阱定向器

可以在 SpectroSERVER 上启用和禁用陷阱定向器。

注意：使用 OneClick 属性编辑器或 CA Spectrum 命令行界面可在服务器的 VNM 模型上设置属性值。

遵循这些步骤:

1. 在 VNM 模型“信息”选项卡中，展开“陷阱管理”子视图。
2. 在“启用陷阱定向器”字段中单击“设置”，然后选择“已启用”。

   将启用陷阱定向器。
3. （可选）要禁用陷阱定向器，请在“启用陷阱定向器”字段中单击“设置”，然后选择“已禁用”。

   将禁用陷阱定向器。

定义缓存记录保留期限

可以控制陷阱定向器陷阱缓存老化的频率。可以定义缓存记录保留期限。

要定义缓存记录保留期限，请为以下属性定义保留期限：

`trap_cache_age_out_minutes (0x12ad5)`

默认值：180（分钟）