

CA Spectrum®

数据库管理指南

版本 9.4



CA Technologies 产品引用

本指南引用了 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>。

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

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

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

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

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

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

本文档由 CA 制作。

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

版权所有 © 2014 CA。保留所有权利。此处涉及的所有商标、商品名称、服务标识和徽标均归其各自公司所有。

目录

第 1 章：概述	7
SpectroSERVER 数据库	7
分布式数据管理器 (DDM) 数据库	7
第 2 章： SpectroSERVER 数据库维护	11
数据库备份	11
联机备份	11
手动备份	16
还原数据库	17
加载数据库	18
初始化并加载数据库	19
加载开发者信息。	20
删除数据库锁定	20
导入和导出模型类型	21
从数据库损坏中恢复	22
SpectroSERVER 数据库故障排除	22
在安装/升级后使用 SSdbload 加载旧对象	23
不适当地使用 SSdbload 的 -i 选项	23
在操作系统/平台之间复制数据库文件	23
建模目录和模型的加载顺序	24
使用相同的开发者 ID 将建模目录对象添加到多个数据库中	24
对数据库文件和目录的 UNIX 文件访问权限	24
CA Spectrum 数据库工具	24
db_remove	25
dbtool	26
MapUpdate	31
HostUpdate	32
reports	33
SSdbload	40
SSdbsave	44
数据库模型转换工具 (DBconv)	46
第 3 章： DDM 数据库维护	55
数据库安全性	55
数据库大小管理	56
数据库备份	56
post_olb_script	57

ddm_save.....	58
ddm_backup	58
有关备份的要点.....	59
数据库还原.....	59
ddm_load	59
有关还原的要点.....	60
数据库维护和优化.....	60
db_maintenance.pl	60
db_optimize.pl	61
DDM 数据库查询.....	62
其他数据库实用工具.....	63
Archive Manager 资源.....	64
DDM 和 Archive Manager 故障排除	65
ARCHMGR.OUT 中的错误消息.....	66
未列出事件.....	67

第 1 章：概述

本章提供 CA Spectrum 数据库维护的概述，并介绍 SpectroSERVER (SS) 和分布式数据管理器 (DDM) 数据库。

本指南从包含多个 SpectroSERVER 的分布式网络环境的角度介绍 CA Spectrum 数据库管理。虽然分布式计算是行业标准，但本指南中的信息也可以适应非分布式环境。CA Spectrum 中的分布式数据库管理包括多个级别的 CA Spectrum 数据库维护。必须遵循各种建模规则和过程，才可以使用大量的应用程序来管理分布式 CA Spectrum 网络及其关联的数据库。

本指南将介绍如何维护每个 SpectroSERVER 的本地数据库以及与多个 SpectroSERVER 关联的历史数据库。

此部分包含以下主题：

[SpectroSERVER 数据库](#) (p. 7)

SpectroSERVER 数据库

位于 $\$SPECROOT/SS$ 目录中的 SpectroSERVER 数据库包含以下信息：

- 建模目录（模型类型和关系），即所有网络信息的结构
- 为该 SpectroSERVER 创建的模型
- 网络事件和统计信息的存储区

代表 OneClick 中网络设备的图标用于报告从实际设备的 *模型*（软件模拟）中检索的信息。模型在 SpectroSERVER 数据库中维护，并使用来自多个源的信息进行更新。每次执行 SpectroSERVER 后保留的信息也会存储在 SpectroSERVER 数据库中。

分布式数据管理器 (DDM) 数据库

每个 SpectroSERVER 都有一个分布式数据管理器 (DDM) 数据库，用于存储可跨多个格局使用的 CA Spectrum 事件和统计信息。然后，CA Spectrum Report Manager 和 CA Spectrum AlarmNotifier 等客户端应用程序可按格局请求、接收和整理数据。DDM 数据库位于 SpectroSERVER $\$SPECROOT/mysql/data/ddmdb$ 目录中。

注意：有关详细信息，请参阅《*分布式 SpectroSERVER 管理员指南*》。

像 SpectroSERVER 数据库一样，将阻止 CA 开发的程序的多个并发用户访问 DDM 数据库。CA Spectrum 应用软锁定文件 (`$SPECROOT/SS/DDM/.DDMDB.LOCK`) 来防止 CA 开发的多个并发应用程序的访问。在某些情况下，例如，在异常关闭后恢复时，可以删除软锁定文件。

重要说明！当 CA 开发的应用程序遇到数据库锁定文件时，会显示一条错误消息向您报警。但是，非 CA 工具和应用程序可能不会检查该锁定，因此不会生成任何消息。如果这些非 CA 实体能够绕过该锁定，则可能会导致数据库损坏。最妥当的做法是，在允许任何非 CA 工具或应用程序访问 CA Spectrum 数据库之前，验证所有 CA Spectrum 进程是否都已关闭。

Archive Manager 控制 DDM 与 SpectroSERVER 数据库之间以及 DDM 数据库与客户端应用程序之间的所有通信。

详细信息：

[删除数据库锁定 \(p. 20\)](#)

Archive Manager

每个格局都有一个 Archive Manager 服务器，该服务器将从 SpectroSERVER 检索事件和统计信息，压缩这些信息，并将其存储在 DDM 数据库中。通过数据压缩可以存储更多的性能数据，并减少应用程序与 DDM 数据库之间的网络通信。

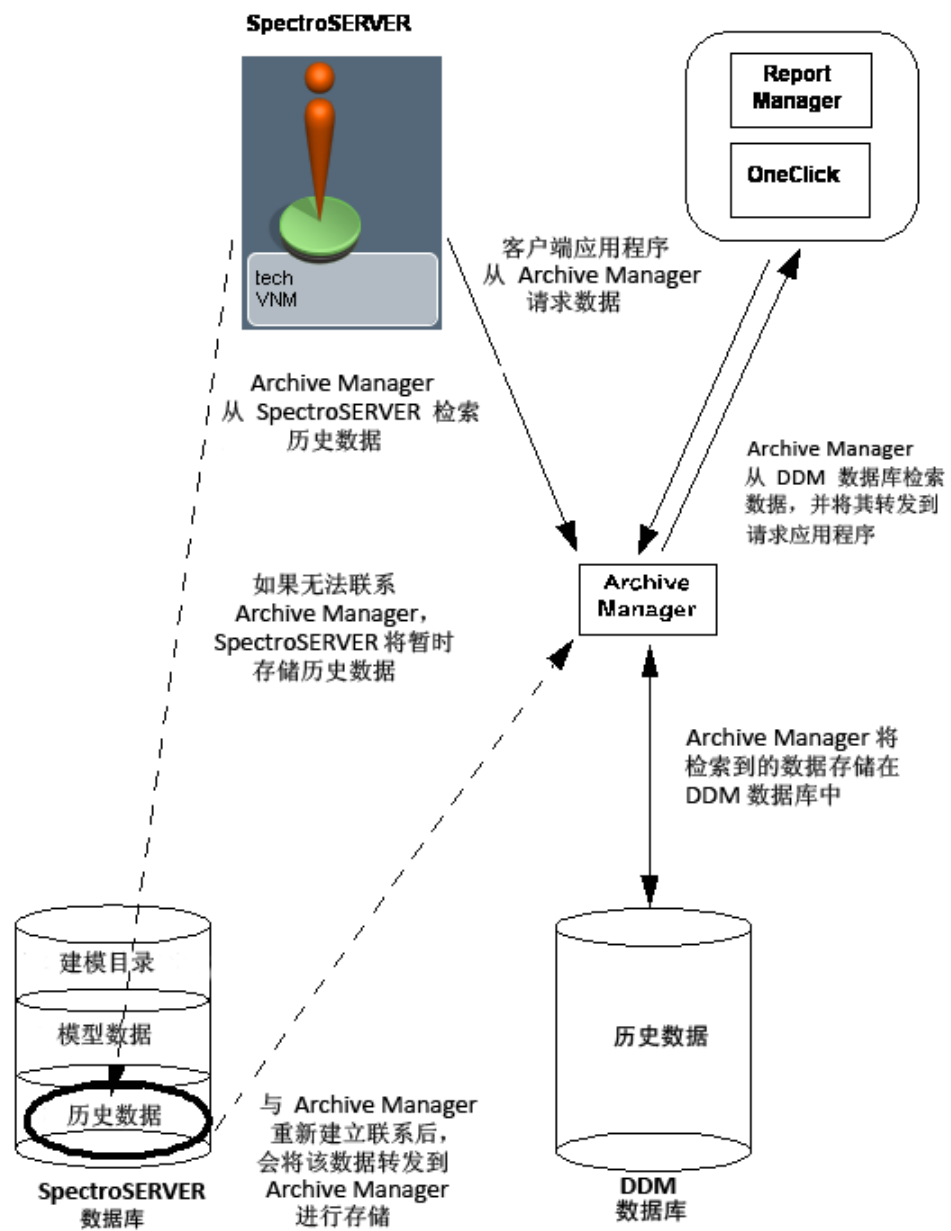
如图所示，如果 SpectroSERVER 无法联系 Archive Manager，SpectroSERVER 将一直存储事件和统计数据，直到重新建立联系。然后，SpectroSERVER 会将数据发送到 Archive Manager 以进行存储。`.vnmrc` 文件中的 Events 和 Statistics Archive 选项确定了 SpectroSERVER 存储的数据量。`.configrc` 文件中的选项确定了历史数据在 DDM 数据库中的存储时间长度。

注意：有关详细信息，请参阅《*分布式 SpectroSERVER 管理员指南*》。

Archive Manager 还可以提供以下信息以响应客户端应用程序的请求：

- 提供了其信息的格局的列表
- 针对每个格局，可用信息的时间范围以及提供了其信息的模型类型的列表
- 针对每个模型类型，提供了其信息的模型的列表
- 针对每个模型，提供了其信息的属性的列表
- 指定时间范围内的统计数据
- 指定时间范围内的事件数据

下图显示了 SpectroSERVER 与 DDM 数据库之间的交互。



第 2 章： SpectroSERVER 数据库维护

本章介绍 SpectroSERVER 数据库的数据库维护过程。分布式数据管理器 (DDM) 数据库的维护过程将在 [DDM 数据库维护](#) (p. 55) 中单独介绍。

此部分包含以下主题：

[数据库备份](#) (p. 11)

[还原数据库](#) (p. 17)

[SpectroSERVER 数据库故障排除](#) (p. 22)

[CA Spectrum 数据库工具](#) (p. 24)

数据库备份

创建数据库的常规备份副本是数据库维护的基础。可靠的数据库备份副本可让您在发生电源故障或其他系统中断后还原数据库。

CA Spectrum 提供两种执行 SpectroSERVER 数据库备份的方法：

- 在运行 SpectroSERVER 的情况下自动执行备份。
- 在关闭 SpectroSERVER 的情况下手动执行备份。

重要说明！ 对于 SpectroSERVER 数据库备份，只支持本部分中介绍的数据库备份方法，即联机备份和使用 `SSdbsave` 实用工具。使用第三方备份软件可能会导致数据库损坏。

联机备份

通过联机备份，您无需关闭 SpectroSERVER 就能创建 SpectroSERVER 数据库的备份副本。根据您的要求，可以按需执行联机备份，也可以排定自动定期执行备份。如果磁盘空间有限，您可以将联机备份配置为自动使用 CA Spectrum `gzip` 实用工具压缩备份文件。

联机备份将保存整个数据库，包括建模目录和模型。但是，联机备份和手动备份操作都不会保存以下信息：

- 缓存的事件信息
- 警报
- 缓存的统计信息
- DDM 数据库中的历史记录（参阅以下注释）
- SpectroSERVER 资源文件 (.vnmrc)

联机备份活动记录为与 VNM 模型关联的事件。CA Spectrum 以警报的形式报告备份操作期间遇到的错误。

联机备份通过以下两个主要步骤执行保存：

1. 生成副本以保留数据库文件的“快照”。

轮询、陷阱处理和网络管理活动在这第一步期间将被暂停。该过程所需的时间相对较短，但作为最佳实践，请考虑排定自动保存操作的频率和时间。执行复制操作的时间取决于工作站硬件和数据库大小。

2. 使用与手动数据库保存实用工具 SSdbsave 相同的格式保存副本（并在需要时压缩副本）。

联机备份本身不会保存 DDM 数据库。但是，您可以将 `post_olb_script` 配置为自动执行，并在每次执行 SpectroSERVER 数据库的联机备份时保存 DDM 数据库。

注意：在容错环境中，请在运行联机备份之前，验证您是否以同一用户身份同时登录到主要和辅助 SpectroSERVER。

详细信息：

[联机备份事件和警报](#) (p. 15)

备份文件维护

当启用了自动备份时，备份文件可能会在备份目录中累积，并耗尽可用磁盘空间。为避免备份失败，请偶尔删除文件，或者将备份文件移到更永久的存储介质。

配置联机备份

在配置联机备份时，您可以指定备份时间间隔以及首次排定备份的日期和时间。如需更高级的排定选项，请使用 `$SPECROOT/SS-Tools` 目录中的 `StartOnlineBackup` 应用程序来启动联机备份。可以通过主机系统的任务排定程序或 `crontab` 启动 `StartOnlineBackup` 应用程序。`StartOnlineBackup` 应用程序中的高级排定选项可以避免当夏令时与排定备份时间有偏差时出现的问题。

遵循这些步骤：

1. 在 OneClick 中，打开“Universe 拓扑”视图，然后选择 VNM 模型。
2. 单击“信息”选项卡，并展开“联机数据库备份”子视图。

3. 根据需要配置以下设置:

自动备份

如果启用该项，将会根据“备份时间间隔”设置中指定的时间间隔使用联机备份功能自动执行联机备份。

备份时间间隔

指定每两次自动备份之间的时间间隔，以小时和分钟为单位。建议使用默认时间间隔 24 小时 0（零）分钟，以便在每天的同一时间备份数据库。输入任意值。例如，168 小时 0 分钟（表示以一周为时间间隔）和 10,080 分钟（同样表示以一周为时间间隔）是等效的。

下一备份的日期及时间

显示下一次排定备份的日期和时间。您可以指定首次备份的日期和时间。但是，后续备份将按“备份时间间隔”设置中指定的时间间隔执行。

备份压缩

如果启用该项，则会先使用压缩实用工具压缩备份文件，然后再将其写入磁盘。保存的压缩文件的文件名附加了后缀 .gz。如果禁用该项，将在不压缩的情况下保存文件。默认值为“已启用”。

备份文件名的前缀

指定备份文件名的用户定义部分。默认前缀为“db_”。但是，您可以指定任意字符串，用于为运行 SpectroSERVER 的系统创建合法文件名。如果未进行设置，将不会在文件名中添加前缀。

文件名后缀指示执行备份的日期和时间，它使用以下格式：

`yyyymmdd_hhmm.SSdb [.gz]`

格式	说明
YYYY	4 位数的年份
MM	月份
dd	月份日期（1 到 31）
hh	一天中的小时 - 1 到 24
mm	分钟 - 00 到 59
gz	指示压缩备份（仅限压缩文件）
SSdb	所有数据库保存文件的默认后缀

备份目录

指定备份文件存储到的目录。建议使用本地目录。默认目录为 `$SPECROOT/SS-DB-Backup`。

最小的必需磁盘空间(MB)

指定启动联机备份必须提供的最小磁盘空间。默认阈值为 20 MB。如果在可用磁盘空间低于该阈值时启动自动备份，系统将生成黄色警报。可能的原因指示了由于磁盘空间不足而导致备份失败。

- 单击“立即开始备份”按钮可以使用当前设置按需启动备份。

联机数据库备份实用工具首先会暂停 SpectroSERVER 操作。然后，它将复制并保存数据库。

注意：在“立即开始备份”按钮旁边将显示联机备份操作进行中的状态。如果发生错误，“事件”选项卡和“警报”选项卡中将显示事件和关联的警报。有关在联机备份期间可能会生成的事件和警报的列表，请参阅 [联机备份事件和警报](#) (p. 15)。

StartOnlineBackup 应用程序

通过 StartOnlineBackup 应用程序，您可以使用操作系统中的排定应用程序来排定定期备份。例如，您可以使用 Windows 环境中的任务排定程序或 Solaris 环境中的 crontab。使用这些应用程序可以避免夏令时与排定备份时间有偏差时存在的潜在问题。此应用程序是使用自动备份设置的更可靠替代措施。

可以通过任务排定程序或 crontab 启动 StartOnlineBackup 应用程序。此实用工具位于 `$SPECROOT/SS-Tools` 目录中。StartOnlineBackup 使用以下语法：

```
StartOnlineBackup -lh <格局句柄>
```

您可以在 VNM 模型的“信息”选项卡上的“联机数据库备份”子视图中设置 StartOnlineBackup 的备份参数。以下参数例外：“自动备份”、“备份时间间隔”、“下一备份的日期及时间”。

详细信息：

[配置联机备份](#) (p. 12)

使用联机备份文件还原数据库

联机备份生成的解压缩备份文件的存储格式与使用具有 `-cm` 选项的 `SSdbsave` 实用工具保存的文件格式相同。

注意：使用 CA Spectrum 随附的 `gzip` 实用工具可以还原带有 `.gz` 后缀的压缩文件。此实用工具会将文件还原到 `SSdbsave` 实用工具使用的格式。

详细信息：

[还原数据库](#) (p. 17)

联机备份事件和警报

以下事件和警报与联机备份关联。

事件代码	事件	事件原因	创建警报
0x00010903	OB_EVENT_GEN_FAILURE	数据库打开失败	红色
0x00010903	OB_EVENT_GEN_FAILURE	数据库关闭失败	红色
0x00010903	OB_EVENT_GEN_FAILURE	导致备份失败的任何其他原因	橙色
0x00010904	OB_EVENT_BACKUP_ON	用户刚刚启用了自动备份	否
0x00010906	OB_EVENT_BACKUP_STARTED	联机备份已启动	否
0x00010907	OB_EVENT_BACKUP_COMPLETED	联机备份已成功完成	否
0x00010908	OB_EVENT_NO_FILE_OR_DIRECTORY	文件/目录不存在，或者没有读取和执行权限	黄色 橙色
0x0001090a	OB_EVENT_NO_CREATE_BACKUP_DIR	无法创建备份目录	橙色
0x0001090b	OB_EVENT_LOW_DISK_SPACE	磁盘空间不足，备份失败	黄色
0x0001090c	OB_EVENT_COPY_FAILED	数据库复制失败	红色
0x0001090e	OB_EVENT_DB_INCONSISTENT	数据库不一致	橙色
0x00010920	OB_EVENT_OFFLINE_SAVE_FAILED	保存过程的脱机部分失败	红色

事件代码	事件	事件原因	创建警报
0x00010921	OB_EVENT_VNM_RESUMED	恢复正常运行，SpectroSERVER 已暂停	否
0x00010922	OB_EVENT_BAD_FILENAME	指定的文件名不是有效的 Posix 目录名称	橙色
0x00010924	OB_EVENT_DBSYNC_FAILED	尝试同步备份服务器失败	橙色

手动备份

您可以使用 CA Spectrum 随附的 `SSdbsave` 数据库实用工具手动创建数据库的备份。数据库包含用于创建模型的模板信息（模型类型、关系和规则）的目录以及模型本身。您可以使用 `SSdbsave` 创建包含其中一个或两个组件的数据库备份副本。建议执行包含建模目录和模型的完整保存。

详细信息：

[SSdbsave](#) (p. 44)

创建完整备份

您可以创建数据库的完整备份副本。完整备份同时包括建模目录（模型类型层次结构、关系和规则）和模型本身。

遵循这些步骤：

1. 导航到 `$SPECROOT/SS` 目录。
2. 确认 SpectroSERVER 或访问 SpectroSERVER 数据库的其他任何程序均未运行。

注意：首次安装后的实用工具位于 `$SPECROOT/SS-Tools` 目录中。如果未在系统路径语句中设置该位置，则您必须使用完整路径、使用相对路径、移动文件，或者将文件链接到 SpectroSERVER 数据库所在的同一目录。

3. 输入以下命令以同时保存建模目录和模型：

```
../SS-Tools/SSdbsave -cm <save_file>
```

保存文件的建议命名方案是在 `save` 命令中包含日期戳和标志。在本例中，您可以使用“c”（目录）和“m”（模型）标志；因此，假设时间为 2013 年 10 月 14 日，请输入 `db_20131014_cm` 作为保存文件名。

随后会创建名为“`db_20001014_cm.SSdb`”的文件。“`.SSdb`”后缀会自动添加。

创建拆分备份

尽管 SpectroSERVER 数据库中的模型每天都可能会发生更改，但是，建模目录只会在您执行升级安装、添加新管理模块或直接使用模型类型编辑器进行更改时才发生更改。因此，如果您的数据库很大，则您可以通过拆分备份来节省时间。在使用拆分备份时，您可以定期保存模型，但只在建模目录发生更改后进行保存。然后，可以将建模目录的备份存储在安全的位置。除了最近保存的模型信息以外，该备份文件还提供了完整的数据库备份。

以下过程介绍如何在 2013 年 10 月 14 日执行数据库的拆分备份，并为建模目录和独立的模型文件各创建一个保存文件。

遵循这些步骤:

1. 导航到 `$SPECROOT/SS` 目录。
2. 确认 SpectroSERVER 或访问 SpectroSERVER 数据库的其他任何程序均未运行。

注意: 首次安装后的实用工具位于 `$SPECROOT/SS-Tools` 目录中。如果未在系统路径语句中设置该位置，则您必须使用完整路径、使用相对路径、移动文件，或者将文件链接到 SpectroSERVER 数据库所在的同一目录。

3. 输入以下命令以仅保存建模目录:

```
../SS-Tools/SSdbsave -c db_20131014_c
```

4. 输入以下命令以仅保存模型:

```
../SS-Tools/SSdbsave -m db_20131014_m
```

建模目录保存在名为 `db_20131014_c.SSdb` 的文件中，模型保存在名为 `db_20131014_m.SSdb` 的文件中。

注意: 可以定期保存模型，也可以在每次对网络模型做出重大更改时保存模型。可以在每次安装新版本的 CA Spectrum、添加新管理模块或者使用模型类型编辑器进行任何目录更改时保存建模目录。

还原数据库

CA Spectrum 数据库由包含模型类型层次结构和关系的建模目录、用于代表网络中实体的模型以及开发者信息组成。通过 `SSdbload` 数据库实用工具程序可以执行以下任务:

- 加载之前备份的数据库文件。
- 初始化数据库，并加载包含建模目录的之前备份的数据库文件。
- 加载开发者信息。

加载数据库

如果未发生数据库损坏，您无需先初始化数据库，就可以使用以下命令来加载备份文件：

```
SSdbload -l
```

您可以使用 `SSdbload` 实用工具将 `SSdbsave` 实用工具创建的文件以及使用 `CA Spectrum` 联机备份工具保存的解压缩文件的内容加载到数据库中。

注意：使用 `CA Spectrum` 随附的 `gzip` 实用工具可以还原带有 `.gz` 后缀的压缩文件。此实用工具会将文件还原到 `SSdbsave` 实用工具使用的格式。

`SSdbload` 将读取保存文件以确定保存的信息类型（建模目录和模型）。然后，它将从数据库中删除所有相应的信息，并将备份信息加载到数据库中。例如，在加载仅包含模型的保存文件（仅使用 `-m` 选项保存）时，`SSdbload` 首先从数据库中删除所有模型信息，然后加载保存文件中的模型信息。

同样，当您加载仅包含目录的保存文件时，该操作将从数据库中删除目录，然后加载保存文件中的目录。如果您创建了单独的备份文件（一个包含建模目录，另一个包含模型信息），则将首先加载目录（模型类型和关系）信息，然后加载模型信息。

重要说明！ 不能使用 `-l` 选项将附加的目录或模型加载到现有数据库中。`-l` 选项无法增补数据库的内容。它用于确定所加载保存文件的内容，并在加载之前删除相同类型的所有现有信息。

遵循这些步骤：

1. 导航到 `$SPECROOT/SS` 目录。
2. 确认 `SpectroSERVER` 或访问 `SpectroSERVER` 数据库的其他任何程序均未运行。
3. 执行 `SSdbload` 命令，并提供要加载到数据库结构中的文件名：

```
../SS-Tools/SSdbload -l <filename>
```

注意：首次安装后的实用工具位于 `$SPECROOT/SS-Tools` 目录中。如果未在系统路径语句中设置该位置，则您必须使用完整路径、使用相对路径、移动文件，或者将文件链接到 `SpectroSERVER` 数据库所在的同一目录。

4. 如果在步骤 3 中加载的备份文件仅包含目录信息，请重复步骤 3，并替换包含模型信息的备份文件的名称。

数据库当前可供使用。

初始化并加载数据库

您可以通过一项操作来初始化并加载数据库。初始化数据库会删除建模目录、模型信息、供应商信息以及未存档的事件和统计日志记录，并刷新所有数据库文件。（要使数据库保持同步，在您初始化 SpectroSERVER 数据库时，请同时使用 `ddm_load` 初始化历史数据库。）

如果您怀疑发生损坏，请始终初始化数据库。在初始化数据库后，`-l` 选项将会根据创建备份文件时使用的 `SSdbsave` 选项加载新的数据库信息（建模目录和模型）。

重要说明！ `-i` 选项将从数据库中删除所有模型、模型类型、关系、供应商信息以及所有未存档的事件和统计日志记录。当使用 `-i` 选项运行 `SSdbload` 时，您可以加载目录以便在没有模型的情况下启动 SpectroSERVER。或者您可以完全还原数据库。要还原数据库，可以同时使用 `SSdbsave -c` 和 `-m` 选项加载某个保存的备份文件，或者加载拆分备份（分别使用 `-c` 和 `-m` 选项保存的文件）。

如果使用拆分备份，请对加载的每个备份文件运行 `SSdbload` 一次。在执行第一个 `SSdbload` 时只使用 `-i` 选项一次。必须在加载模型之前加载目录。

要通过一项操作同时进行初始化和加载，请将 `-i` 选项与 `-l` 选项结合使用。在结合 `SSdbsave` 使用 `-c` 和 `-m` 选项从保存的文件中加载新数据库信息时，建议使用此方法。

遵循这些步骤:

1. 导航到包含数据库的目录。
2. 确认 SpectroSERVER 或访问 SpectroSERVER 数据库的其他任何程序均未运行。
3. 在初始化数据库时，如果使用比当前可执行文件更旧的 `.after` 文件，请小心操作。如果 `.after` 文件比可执行文件更旧，可能不会显示预期的信息。有关 `.after` 文件的说明，请参考 `SSdbload`。
4. 输入以下 `SSdbload`:

```
../SS-Tools/SSdbload -i -l <filename>
```

数据库将被初始化，并连同指定备份文件中的信息一起加载。然后，数据库便可供使用。

详细信息:

[SSdbload](#) (p. 40)

加载开发者信息。

允许注册的开发者使用具有 `-d` 选项的 `SSdbload` 将开发者信息（包含开发者 ID 的文件）加载到 SpectroSERVER 数据库中。在执行任何数据库初始化操作后，必须重新加载开发者信息，但是，每次初始化之后，可以只加载此信息一次。尝试使用 `-d` 选项执行第二个 `SSdbload` 会生成错误消息，并保持开发者信息不变。

重要说明！ 只能将开发者信息加载到一个数据库中，并对该数据库执行所有的建模目录编辑。然后，可以根据需要使用 `SSdbload` 将修改传播到其他数据库。但是，在多个数据库中加载相同的开发者信息可能会导致模型类型、属性或关系句柄重复。如果出现重复，将无法跨格局关联历史数据。

遵循这些步骤：

1. 导航到 `$SPECROOT/SS` 目录。
2. 确认 SpectroSERVER 或访问 SpectroSERVER 数据库的其他任何程序均未运行。

注意：首次安装时，`SSdbsave` 位于 `$SPECROOT/SS-Tools` 目录中。如果未在系统路径语句中设置其位置，则您必须使用完整路径、相对路径、移动它，或者将它链接到 SpectroSERVER 数据库所在的同一目录。

3. 输入以下命令：

```
../SS-Tools/SSdbload -d <开发者文件名>
```

删除数据库锁定

每当 CA 开发的程序访问 CA Spectrum 数据库时，便将在该数据库目录中创建一个锁定文件。SpectroSERVER 数据库中的锁定文件名为 `.VNMDB.LOCK`。DDM 数据库中的锁定文件名为 `.DDMDB.LOCK`。锁定文件充当数据库当前使用的指示器，无法由 CA 开发的其他程序访问。

CA 开发的以下程序可创建锁定文件：

- converter
- dbtool
- lh_set
- mte (模型类型编辑器)
- reports
- SpectroSERVER
- SSdbdelete

- SSdbload
- SSdbsave

当 CA 开发的程序遇到正常终止时，将删除锁定文件。但是，CA 开发的程序偶尔会异常终止，此时将保留锁定文件。如果未删除锁定文件，锁定文件将会阻止执行 CA 开发的其他需要访问数据库的程序。如果锁定文件存在并有另一 CA 开发的程序启动了，将显示以下消息：

数据库已被锁定：<用户 ID>，
进程：<进程名称>，
进程 ID：<进程 ID>，
位于网络节点：<节点名称>，
启动时间：<日期/时间戳>

对于您确信是由某个异常终止的程序留下的 .VNMDB.LOCK 或 .DDMDB.LOCK 文件，您可以手动将其删除，但首先应该确认该程序未运行。如果该程序仍在运行（并在阻止更高优先级的程序访问数据库），请先尝试正常终止该程序。如果无法做到这一点，则可以使用 UNIX kill 命令（不带选项）或 Windows 任务管理器中的“结束进程”按钮来停止该程序。其中的任何一种方法都可以删除锁定文件。

如果该程序异常终止，您可以根据情况导航到 `$SPECROOT/SS` 或 `$SPECROOT/SS/DDM` 目录，并输入以下命令之一来手动删除锁定文件：

- `rm .VNMDB.LOCK`（用于从 SS 目录中删除锁定）
- `rm .DDMDB.LOCK`（用于从 DDM 目录中删除锁定）

重要说明！ 为了保持数据库的完整性，当您被迫从 `$SPECROOT/SS` 或 `$SPECROOT/SS/DDM` 目录中删除锁定文件时，必须同时还原 SpectroSERVER 和 DDM 数据库。在您执行数据库还原操作之前，SpectroSERVER 不会重新启动。

另请记住，.VNMDB.LOCK 和 .DDMDB.LOCK 文件不会阻止非 CA 开发的程序访问数据库，当此类程序遇到这些锁定时，不会生成错误消息。因此，在允许任何非 CA 工具或应用程序访问 CA Spectrum 数据库之前，应确保关闭访问该数据库的所有 CA Spectrum 进程。

详细信息：

[还原数据库](#) (p. 17)

导入和导出模型类型

您可以使用 dbtool 实用工具从 SpectroSERVER 数据库导入和导出模型类型及关联的对象（属性、关系和元规则）。

详细信息:

[dbtool](#) (p. 26)

从数据库损坏中恢复

在某些情况下（包括硬件故障和电源中断），可能会导致数据库损坏并生成如下所示的错误消息：

```
9 月 18 日, 15:42:39, CsSSDbRp.cc(642) 上出错:  
表打开失败: @ TableImpl.cc:674(0x2)  
Db::open: 不存在该文件或目录  
无法打开数据库。正在退出 VNM。  
格局未初始化。正在退出 VNM。
```

当 SpectroSERVER 或其他应用程序生成不直接构成特定原因的错误消息（如缺少用户模型、存在数据库锁定等）时，也可能发生数据库损坏。

如果您保留了定期备份的排定，则通常只需使用 **SSdbload** 实用工具初始化数据库并重新加载数据库上次的已知“正确的保存版本”（即包括建模目录和模型，并在出现任何损坏迹象之前创建的版本），就能从数据库损坏中恢复，且只会丢失少量的信息。如果您没有已知的正确数据库保存版本，或者在重新加载确信正确的保存版本之后仍然出现应用程序错误，请联系您的 CA 支持代表。

详细信息:

[SSdbload](#) (p. 40)

SpectroSERVER 数据库故障排除

本节介绍客户曾经报告给 CA 支持部门的某些常见问题。

在安装/升级后使用 SSdbload 加载旧对象

在安装或升级后，使用 SSdbload 加载旧的模型、模型类型或关系可能会导致严重的数据库问题。避免这些问题的最佳途径是了解 SSdbload 的工作原理。当您尝试使用 -l 选项加载某个文件时，SSdbload 将检查该文件以确定它包含的对象类型（取决于创建该文件时与 SSdbsave 结合使用的选项标志）。对于该文件中的每个对象类型（建模目录和/或模型），SSdbload 首先从数据库中删除该类型的所有现有对象。例如，如果执行：

```
SSdbload -l somefile_c.SSdb
```

以使用旧建模目录更新最新升级的数据库，则会将数据库中的建模目录替换为保存文件中的某个建模目录。这可能导致数据库不可用，因为可能需要使用升级后的建模目录来操作新版本的 CA Spectrum。此外，对模型类型编辑器所做的任何修改将会丢失。

不适当地使用 SSdbload 的 -i 选项

使用 SSdbload 工具的 -i 选项可能会造成严重的后果。使用此选项时，将删除数据库中的所有现有对象。删除的对象包括模型类型、模型、关系、属性、规则、未存档的事件和未存档的统计信息。因此，如果执行以下命令：

```
SSdbload -i -l somefile_m.SSdb
```

并且 somefile_m.SSdb 文件只包含模型，则加载将会失败。如果使用仅包含建模目录（即只使用 -c 选项保存）的文件执行同一条命令，则生成的数据库仅包含从该文件加载的模型类型和关系。数据库中先前存在的所有其他对象（包括用户创建的所有模型）都将丢失。

在操作系统/平台之间复制数据库文件

数据库文件 (*.db、*.ix、log.*) 特定于操作系统和平台。不支持将一个操作系统或平台中的数据库复制到另一个操作系统或平台。移动这些文件中包含的数据的唯一受支持方法是在源计算机上运行 SSdbsave，在目标计算机上运行 SSdbload。只要这些工具在每台计算机上都存在，并且它们是针对各自的服务器编译的，那么，数据传输就会成功。

建模目录和模型的加载顺序

了解有关 CA Spectrum 模型的两个事实非常重要：

- 模型是模型类型的实例，后者在建模目录中定义。
- 模型与模型类型之间具有依赖关联关系。

如果给定模型的模型类型不存在或版本不正确，则尝试加载包含该模型的文件将会失败。始终要先加载所有必备的建模目录，然后再加载从该建模目录创建的模型。

使用相同的开发者 ID 将建模目录对象添加到多个数据库中

当使用模型类型编辑器将建模目录对象（模型类型、关系和属性）添加到数据库时，将为这些对象分配唯一句柄。句柄包括活动的开发者 ID 以及该对象类型的下一个可用顺序号。但是，每个数据库将单独分配并维护这些顺序号。因此，如果使用相同的开发者 ID 将新对象添加到其他数据库，则相同的句柄可能代表每个数据库中的不同对象。在分布式 SpectroSERVER 环境中，如果存在这种冲突，将无法跨格局关联历史数据。

为了避免句柄重复的问题，请只修改一个数据库中的建模目录信息。使用具有 -c 选项的 SSdbsave 保存更改，然后根据需要具有 -l 选项的 SSdbload 将更改传播到其他数据库。

对数据库文件和目录的 UNIX 文件访问权限

要正常运行 SpectroSERVER 和数据库工具，您必须拥有对所有数据库文件（*.db、*.ix、log.*）以及包含这些文件的目录的写入权限。如果您从某个数据库工具或 SpectroSERVER 接收到表示程序“无法打开锁定文件”的消息，请检查对数据库文件和目录的权限。

CA Spectrum 数据库工具

本节介绍以下数据库工具、实用工具程序或脚本：

- db_remove
- dbtool
- HostUpdate
- MapUpdate
- reports
- SSdbload

- SSdbsave
- 数据库模型转换工具 (DBconv)

注意：首次安装时，上面列出的所有工具都位于 `$SPECROOT/SS-Tools` 目录中。如果未在系统路径语句中设置某个工具的位置，请使用完整路径，或者将其移至 SpectroSERVER 数据库所在的同一目录。

db_remove

该实用工具用于从 SpectroSERVER 数据库的建模目录中删除弃用的模型类型（及其关联的原始属性和元规则）。

注意：此工具不适用于常规用途。除非您收到具体的指示或者 CA 发布的技术公告要求您这么做，否则不要运行 `db_remove`。有关详细信息，请联系 CA 支持代表。

该实用工具采用以下格式：

```
db_remove [-debug] [<MTH_FILE>]
```

-debug

启用详细的信息消息输出。

<MTH_FILE>

指定文本文件，该文件包含要从数据库中删除的模型类型句柄的空格分隔列表。

错误

警告: 模型类型 0x???????? - 无法删除 “模型类型名称”。

在偶尔情况下，当您（或者其他供应商或合作伙伴）从所删除的模型类型中派生新模型类型时，就会看到此消息。在这种情况下，请运行模型类型编辑器以查看从此模型类型派生的内容。然后，将派生的一个或多个模型类型移到其他未弃用的派生点。最后，对原始 MTH_FILE 重新运行 `db_remove` 工具。

警告: 模型类型 0x???????? - 当前已在默认属性值中引用 “模型类型名称”。该引用出现在属性 0x???????? 中 - “属性名称”，位于模型类型 0x???????? - “模型类型名称”。

如果您（或者其他供应商或合作伙伴）添加了从某个属性（`MODELTYPE_HANDLE` 或 `MODELTYPE_HANDLE` 列表）的默认值到所删除模型类型句柄值的引用，就会看见此消息。此消息不会阻止删除模型类型。但是，应调查属性的默认值。请根据属性的使用方式，删除引用或者将它替换为适当的未弃用模型类型句柄。

dbtool

随着网络的扩充，您通常会向数据库添加模型类型。在某些情况下，您可能想要添加新的模型类型但不安装新数据库。可以使用 **dbtool** 实用工具来完成以下相关任务：

- 从 SpectroSERVER 数据库中的永久目录中导出模型类型。
- 将模型类型导入 SpectroSERVER 数据库中的永久目录。
- 显示（转储）导出文件的内容。

转储功能会将输出发送到工作站的标准输出（通常为显示屏）。但是，也可以将输出发送到文件或打印机。

模型类型编辑器包括类似的功能用于导出和导入模型类型。但是，**dbtool** 实用工具允许您将多个文件指定为命令行参数。因此，它更适用于对一组文件进行批处理。

注意： 有关详细信息，请参阅《*模型类型编辑器用户指南*》。

dbtool 实用工具根据功能（导出、导入或转储）使用下列文件类型之一：

- 带有文件扩展名 **.m** 的模型类型列表文件。此文件包含模型类型 ID 代码的列表。每个 ID 显示在单独的行（以回车符分隔）中，或以空格彼此分隔在同一行中。您可以使用喜爱的 **shell** 文本编辑器创建这些文件。
- 带有文件扩展名 **.e** 的提取文件。您可以使用 **dbtool** 导出功能或使用模型类型编辑器创建这些二进制文件（又称为目录文件）。它们包含数据库中的，与关联的 ***.m** 文件（如果由 **dbtool** 生成）中所列模型类型相关的所有信息。或者，如果目录文件是由模型类型编辑器生成的，则这些文件将包含与所有选定模型类型相关的信息。

您可以通过目录文件使用介质（例如电子邮件文件、DVD 等）将模型类型从一个系统传输到另一个系统。

重要说明！ 在您运行 **dbtool** 之前，请关闭 SpectroSERVER 以及访问 SpectroSERVER 数据库的任何其他程序，包括第三程序。始终从包含 SpectroSERVER 数据库的目录运行 **dbtool**。

dbtool 实用工具将加载数据库，以便在可能的情况下使用符号名称。

此实用工具采用以下格式：

```
dbtool      [dump <file>.e [ <file>.e ...] ]
dbtool      [dump_mt <file>.e [ <file>.e ...] ]
dbtool      [import <file>.e [ <file>.e ...] .xml [ <file>.xml ...] ]
dbtool      [export <file>.m [ <file>.m ...] ]
```

dump

以可读的形式显示指定 .e 文件（目录文件）的内容。与 dump_mt 参数不同，此参数不仅在输出中包括模型类型信息，而且还包括属性、关系和元规则信息。如果需要，您可以指定多个要转储的目录文件。

注意：此选项只会报告源自所导出模型类型的那些属性的属性名称。它并不提供有关扩展标志、OID 前缀值或 OID 参考值的更改的输出。

此参数对数据库不起作用。

dump_mt

以可读的形式显示指定 .e 文件中所列模型类型的相关信息。如果需要，您可以指定多个要转储的目录文件。与 dump 参数不同，此参数不会在输出中包括属性、关系或元规则信息。

此参数对数据库不起作用。

导入

将指定 .e 或 .xml 文件中的模型类型、属性、关系和元规则导入数据库。如果需要，您可以指定多个要导入的目录文件。

导入已在数据库中存在的模型类型将生成有关将重新定义的模型类型的警告消息。可以忽略此警告消息。

export

使用指定的 .m 文件创建一个目录文件，该目录文件包含 .m 文件中的模型句柄指定的模型类型。您可以指定多个 .m 文件以创建多个目录文件。

.m 文件中的模型类型句柄条目必须是以“0x”（零后接小写“x”）开头的十六进制整数。此外，必须至少使用一个空格字符或换行符分隔各个条目。

help

显示有关命令的用法信息。

详细信息：

[将目录文件的内容定向到输出设备 \(p. 30\)](#)

使用 dbtool 导出模型类型

您可以使用 `dbtool` 实用工具导出模型类型。

注意：您只能导出使用 SpectroSERVER 数据库中当前加载的开发者 ID 创建的模型类型、属性和关系（以及关联的元规则）。该开发者 ID 是这些对象的“所有者”。如果您不是导出操作中包含的某个模型类型或其他对象的所有者，则操作将会终止并显示错误消息。

导出过程生成的目录文件（.e 文件）包含以下信息：

- 源自所导出模型类型的属性描述符。
- 已专用化的属性描述符（例如，通过指定默认值来覆盖继承的值）。
- 模型类型 and 任何上级模型类型以先例或断言形式参与的关系及关联的元规则。

“边缘”模型类型至少有一个未在相同 *.e 文件中导出的基本模型类型。这些类型的不同之处在于，它们包含从未包括的基本模型类型中继承的属性值和扩展名。这种包含可以确保这些值和扩展名的可用性。

遵循这些步骤：

1. 将工作目录更改为包含数据库的目录。
2. 确认 `dbtool` 实用工具不在当前目录或系统搜索路径中。或者，当您稍后调用 `dbtool` 命令（步骤 4）时，请使用适当的相对或绝对路径名。
3. [创建 .m 文件](#) (p. 28) 用于指定要导出的模型类型。
4. 使用以下命令导出模型类型：

```
../SS-Tools/dbtool [export<filename_1>.m [<filename_2>.m ... ] ]
```

以下命令用作示例：

```
../SS-Tools/dbtool export smart_hub.m smart_router.m
```

此示例中的命令将导出 `smart_hub.m` 中定义的模型类型，并将生成的输出写入名为 `smart_hub.e` 的文件。然后，它以相同的方式处理 `smart_router.m`。

创建 .m 文件

`dbtool` 实用工具的导出功能使用一个或多个 *.m 文件来指定模型类型句柄导出的模型类型。在使用 `dbtool` 运行导出之前，请使用您喜好的 shell 文本编辑器创建其中的一个或多个文件。

通常，要导出的模型类型列表包括所导出的模型类型所需的，并且不在目标数据库中存在的任何基本模型类型。但是，依存关系通常限制为一个或多个“核心”目录中包含的某些常用基本模型类型。这些目录包含为基本 CA Spectrum 系统的一部分。

遵循这些步骤:

1. 以“0x”开头（零后接小写“x”）的十六进制整数形式输入模型句柄条目。0x 应该后接 8 位数，依次为 4 位数开发者 ID 和 4 位数顺序号。
2. 应至少使用一个空格字符或换行符分隔各个模型句柄条目。
3. 使用 .m 文件扩展名命名列出句柄的文件。

例如，以下列表将指定要导出的五个模型类型（使用默认的开发者 ID 创建）：

```
0xffff0003 0xffff0008  
0xffff0017 0xffff0023  
0xffff0045
```

此示例将生成一个 *.e 文件，其中包含使用当前加载的开发者 ID（在本例中为默认开发者 ID）创建的第 3、8、17、23 和 45 个模型类型的数据库信息。

重要说明！ 必须限制为一次只能由一个应用程序访问数据库。当使用 dbtool 时，必须拒绝所有其他应用程序（包括 OneClick 和模型类型编辑器）访问。当 CA 开发的程序自动锁定其他 CA 产品时，如果第三方应用程序绕过此预防措施，则可能会发生数据库损坏。

使用 dbtool 导入模型类型

您可以使用 dbtool 导入功能将模型类型从您先前使用 dbtool 导出功能或使用模型类型编辑器创建的一个或多个目录文件（.e 文件）导入到 SpectroSERVER 数据库中。

必须限制为一次只能由一个应用程序访问数据库。当使用 dbtool 时，必须拒绝所有其他应用程序（包括 OneClick 和模型类型编辑器）访问。当 CA 开发的程序自动锁定其他 CA 产品时，如果通过第三方应用程序忽略或绕过此预防措施，则可能会发生数据库损坏。

遵循这些步骤:

1. 备份 SpectroSERVER 数据库。
2. 将当前工作目录更改为包含您的数据库的目录。
3. 确认要导入的模型类型文件不在当前目录或系统搜索路径中。

- 如果您的数据库已初始化（存在带 .d 和 .k 扩展名的文件），请继续执行下一步。否则，请使用 `SSdbload -i -l` 实用工具命令初始化数据库并加载核心模型类型派生。
- 使用以下命令导入模型类型：

```
../SS-Tools/dbtool import [<filename_1.e> ... <filename_n.e>]
```

以下命令用作示例：

```
../SS-Tools/dbtool import rmon1.e rmon2.e
```

在导入最后一个源文件的内容后，将显示一条消息来指示操作已完成。

将目录文件的内容定向到输出设备

当您使用 `dump` 参数或 `dump_mt` 参数运行 `dbtool` 实用工具时，该实用工具会将输出发送到工作站的标准输出设备（通常是显示屏）。如果您想要将输出直接重定向到文件或打印机，请根据需要在命令行中提供标准的 UNIX 管道命令来重定向输出。

示例 1

以下命令会将名为 `rmon.e` 的目录文件的输出转储到工作站的标准输出设备：

```
dbtool dump rmon.e
```

输出将作为一次性显示内容出现在工作站上，如果输出内容太长，您只能看到输出的最后部分（即，显示器能够显示的最多文本行数）。

示例 2

以下命令将转储与前一示例中相同的目录文件的输出，但是，会将输出以增量显示文件的形式发送到工作站屏幕，并显示第一个屏幕：

```
dbtool dump rmon.e | more
```

在查看屏幕时要逐行浏览，请按回车键；要切换到下一屏幕，请按空格键。

示例 3

以下命令将转储与前一示例中相同的目录文件的输出，但是，会将输出写入到 ASCII 文件名 `filesave.out`：

```
dbtool dump rmon.e > dumpouts/filesave.txt
```

该文件将在 `$SPECROOT/SS/dumpouts` 目录中创建，您必须事先创建了该目录。

示例 4

以下命令将转储与前一示例中相同的目录文件的输出，但是，输出将以打印文件的形式发送到 <ptr> 指定的打印机：

```
dbtool dump rmon.e > lpr -P<ptr>
```

排除 dbtool 故障

在运行 dbtool 时，您可能会遇到以下错误：

数据库打开失败

SpectroSERVER 数据库不在当前目录中。

数据库文件缺少读取和/或写入权限

您的开发者 ID 未加载到数据库中，或者您不是指定数据库的有效用户。

数据库已被锁定: <用户 ID>

进程: <进程名称>

进程 ID: <进程 ID>

位于网络节点: <节点名称>

启动时间: <日期/时间戳>

数据库由其他进程锁定。

详细信息：

[加载开发者信息。](#) (p. 20)

[删除数据库锁定](#) (p. 20)

MapUpdate

MapUpdate 是用于修改和显示格局映射的实用工具。该程序位于 `$SPECROOT/SS-Tools` 目录中，可用于执行以下任务：

- 从格局映射中删除格局条目
- 显示当前的格局映射

如果您要删除辅助 SpectroSERVER，请确保运行 MapUpdate 以从主要 SpectroSERVER 上已加载的格局列表中删除辅助 SpectroSERVER。如果您正在使用格局条目的超时值，请在辅助 SpectroSERVER 上的格局条目超时之前运行 MapUpdate -remove。否则，您可能无法正常地从主要 SpectroSERVER 的已加载格局列表中删除辅助 SpectroSERVER。

注意：在以前的 CA Spectrum 版本中，默认情况下，格局条目在一个小时后超时，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。您必须使用 MapUpdate 手动从格局映射中删除条目。您可以使用格局条目的超时值。有关详细信息，请参阅《分布式 SpectroSERVER 管理员指南》。

该命令采用以下格式：

```
MapUpdate [-remove 格局句柄] [-precedence 优先级] [-view]
```

-remove 格局句柄

指定要删除的格局的句柄。必须先关闭要删除的 SpectroSERVER。此外，如果您要删除辅助 SpectroSERVER，主要 SpectroSERVER 必须正在运行。

默认值： 0x400000

-precedence 优先级

要删除的格局的优先级值。

-view

显示当前的格局映射（如果已提供）。

HostUpdate

HostUpdate 实用工具位于 SS-Tools 目录中，可让您删除所指定主机的，已按主机分区的所有格局映射条目。

```
HostUpdate [-remove 主机名] [-view]
```

-remove

删除您通过输入主机名指定的主机的所有条目。

注意：格局映射中的主机条目将自动超时。因此，此选项可能是不必要的。但是，如果自动超时机制失败，或者您要在超时间隔之前删除条目，则可以使用此选项。

-view

显示已按主机分区的所有条目。

reports

该命令行实用工具位于 `SS-Tools` 目录中，可用来显示当前建模目录中的选定对象的列表。从 `SpectroSERVER` 数据库的安装目录运行 `reports`。

该命令将在报告生成序列期间锁定数据库。因此，您一次只能运行一份报告。

注意：在执行 `reports` 之前，必须关闭 `SpectroSERVER` 以及访问 `SpectroSERVER` 数据库的任何其他程序，例如 `VNM`、模型类型编辑器或任何第三方实用工具。否则，可能会发生数据库损坏。

该实用工具采用以下格式：

```
reports [-mtype <名称模式>] [-relation <名称模式>] [-handle <句柄>] [-attrflags
<defglmprsvw>] [-fields <cdefgimnotvGE>] [-types <bierdtcgmRlaoIAOTU>]
[-recursive]
[-invisible] [-lists] [-nolists] [-groups] [-help]
```

-mtype

指定要包括在报告中的模型类型。将包括名称中含有指定文本字符串的所有模型类型。例如，如果模型类型是 `IRM`，则报告将列出 `Hub_CSI_CIRM`、`Hub_CSI_IRM2`、`Hub_CSI_SIRM` 以及名称中含有“`IRM`”字符串的所有其他模型类型的部分。

注意：对于“所有适用的”条目，您可以使用通配符替换句点 (`.`)。请谨慎地使用通配符选项，因为这需要很长时间才能显示报告。从 `$SPECROOT/SS` 目录运行以下命令，以显示有关不带属性信息的所有模型类型的报告：

```
../SS-Tools/reports -mtype . -fields e
```

-relation

指定要包括在报告中的关系。输出将列出一个模型类型与另一个模型类型的关联方式。

注意：对于“所有适用的”条目，您可以使用通配符替换句点 (`.`)。从 `$SPECROOT/SS` 目录运行以下命令，以显示有关所有关系的报告：

```
../SS-Tools/reports -rel .
```

-handle

指定要包括在报告中的模型类型的十六进制句柄, 使用或不使用前置的“0x”前缀。将包括句柄中含有指定文本字符串的所有模型类型。此参数类似于 `mtype` 参数, 不过它接受模型类型句柄而不接受名称字符串。

例如, 如果指定的句柄是 `0x180027`, 则报告将包括在 `0x180000` 开发者 ID 下创建的第 27 个模型类型的部分。相比之下, 如果句柄是 `180`, 则报告将不仅包括在该开发者 ID 下创建的所有模型类型, 而且还包括句柄中任意位置包含这三个数字的任何其他模型类型。

-attrflags defglmprsvw

筛选报告, 以便仅在报告中包括具有某个指定标志集的模型类型的属性。

d

数据库

e

外部

f

全局

g

已保证

l

已记录

m

内存

p

已轮询

r

可读

s

已共享

v

保留

w

可写

注意：连续输入两个单引号或双引号字符来代替标志列表，相当于输入了所有标志。如果您包含此参数但不指定值，则 **reports** 实用工具将会失败并返回错误。将显示帮助文件。有关这些属性组件的详细说明，请参阅《模型类型编辑器用户指南》。

-fields cdefgimnotvGE

筛选报告以仅包括指定的属性组件（字段）。

c

创建器模型类型

d

OID 引用

e

不打印属性信息

f

标志

g

轮询组

i

属性 ID

m

清单常量名称

n

带有开发者 ID 的名称

o

OID 前缀

t

类型

v

默认值

G

组 ID

E

枚举值列表

注意：连续输入两个单引号或双引号字符来代替标志列表，相当于输入了所有标志。如果您包含此参数但不指定值，则 **reports** 实用工具将会失败并返回错误。将显示帮助文件。有关这些属性组件的详细说明，请参阅《模型类型编辑器用户指南》。

-types bierdtcgmMRLazoiAGOTU

筛选报告以仅包括指定类型的属性：

b

布尔值

i

整数

e

枚举

r

实数

d

日期

t

时间

c

计数器

g

标尺

m

模型句柄

M

模型类型句柄

R

关系句柄

I	格局句柄
a	属性 ID
z	文本字符串
o	对象 ID
I	IP 地址
A	代理 ID
O	八进制字符串
T	标记的八进制数
U	64 位无符号整数

-recursive

在报告中包括降序（子级）模型类型。

-invisible

在报告中包括以下模型类型：

- 与搜索模式匹配的所有可视模型类型（可视模型类型标志设置为 **true**）
- 通过当前加载的开发者 ID 创建的，与搜索模式匹配的所有不可见模型类型（可视模型类型标志设置为 **false**）。

-lists

仅在报告中包括允许多个值的属性。

-nolists

不在报告中包括允许多个值的属性。

-groups

在报告中包括模型类型组。

-help

显示有关命令的用法信息。

详细信息：

[运行模型类型报告](#) (p. 38)

运行模型类型报告

模型类型报告列出特定模型类型的属性信息。要运行模型类型报告，请使用以下语法：

```
../SS-Tools/ \  
reports [-mtype <model type>][-attrflags w][-fields][-invisible]
```

或者

```
../SS-Tools/ \  
reports [-handle <model handle>][-attrflags ""][-fields][-invisible]
```

例如，以下命令将生成有关 HUB_CSI_IRM2 模型类型的报告：

```
reports -mtype HUB_CSI_IRM2 -attrflags e -fields don -invisible
```

该报告列出具有外部标志集的所有属性。对于包括的每个属性，它将标识 OID 引用、OID 前缀和名称（带有开发者 ID）。该报告还包括指定模型类型的基本模型类型（父模型类型）的完整列表；基本模型类型按继承顺序列出。

该报告将发送到工作站的标准输出设备。

详细信息：

[将报告定向到输出设备](#) (p. 38)

将报告定向到输出设备

`reports` 实用工具会将输出发送到工作站的标准输出设备（通常为显示屏）。要将输出直接重定向到文件或打印机，请在命令行中提供标准的 UNIX 管道命令。

示例 1

以下命令将生成有关 HUB_CSI_IRBM 模型类型的报告：

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n
```

该报告包括标准的标题部分，用于标识模型类型开发者、其名称和句柄、六个属性标志的状态以及直接基本模型类型的标识。报告的剩余部分限制为带有外部标记的属性，报告中的行项只列出了属性名称。

该报告将以仅一次性显示的形式出现在工作站屏幕上。如果报告太长，您只能看到报告的最后部分（即，显示器能够显示的最多文本行数）。

示例 2

以下命令将生成相同的报告，但是，会将输出以增量显示文件的形式发送到工作站屏幕，并显示第一个屏幕：

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n | more
```

在查看屏幕时可以按回车键逐行浏览；按空格键可查看下一屏幕。

示例 3

以下命令将生成相同的报告，并将报告写入到当前目录中名为 REPORT_1 的 ASCII 文件：

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n > REPORT_1
```

如果命名的文件在当前目录中存在，它将被新报告覆盖。

您可以在报告文件名的前面添加目录路径。

示例 4

以下命令将生成相同的报告，并将其作为打印文件发送到 <ptr> 指定的打印机：

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n > lpr -P<ptr>
```

运行关系报告

关系报告根据选定的关系列出一个模型类型与另一个模型类型的关联方式。要运行关系报告，请使用以下语法：

```
../SS-Tools/reports -rel <relation>
```

其中，<relation> 是要包括在报告中的关系的名称。一次指定一种关系。

例如，以下命令将生成有关 Connects_to 关系的报告：

```
../SS-Tools/reports -rel Connects_to
```

该报告将列出 Connects_to 关系中的每个有序规则，系统会将该报告发送到工作站的标准输出设备。有关将报告写入文件或者将报告发送到打印机的信息，请参阅[将报告定向到输出设备](#) (p. 38)。

详细信息：

[将报告定向到输出设备](#) (p. 38)

SSdbload

此实用工具程序位于 $\$SPECROOT/SS-Tools$ 目录中，可让您使用先前创建的备份文件还原 SpectroSERVER 数据库，加载开发者 ID 信息，或者为容错环境中的 SpectroSERVER 设置优先级值。

注意：有关建立容错的详细信息，请参阅《*分布式 SpectroSERVER 管理员指南*》。

在执行 SSdbload 之前，必须关闭 SpectroSERVER 以及访问 SpectroSERVER 数据库的任何其他程序。

该实用工具采用以下格式：

```
SSdbload [-quiet] [-initialize] [-developer <DEV_INFO_FILE>] [-load] [-models]
[-catalog] [-replace <PRECEDENCE>] [-add <PRECEDENCE>] [-port <PORT_NO>]
[-showmap] [-version]
[-extension] [-new_primary <NEWHOSTNAME>]
[<SAVE_FILE>]
```

其中，参数名称的第一个字母以粗体显示，您只需使用该字母，而不必输入整个字符串。

-quiet

禁用提示（交互模式）。用于从脚本内部运行加载命令。

-initialize

通过删除建模目录、所有模型以及所有未存档的事件和统计日志记录来初始化数据库。

重要说明！如果您使用 **-i**（初始化）选项，则至少需要还原目录。如果使用拆分备份，则必须为加载的每个备份文件运行 SSdbload 一次。**-i** 选项只能使用一次，具体而言，在首次执行 SSdbload 时使用。在加载模型之前加载建模目录。

-developer

加载您使用 <DEV_INFO_FILE> 变量指定的开发者信息文件。

注意：只能将开发者信息加载到一个数据库中，并对该数据库执行所有的建模目录编辑。对多个数据库中使用相同的开发者信息可能会导致模型类型、属性或关系句柄重复。

可以使用 -d 选项来加载开发者信息，但只能使用一次。尝试使用 -d 选项执行第二个 SSdbload 会生成错误消息，并保持开发者信息不变。

-load

从您使用 <SAVE_FILE> 变量指定的保存文件中加载包含对象的数据库。

重要说明！ -l（加载）选项不可用于将模型或建模目录组件添加到现有数据库中。它不会增补数据库的内容，而是先删除保存文件中所包含的相同类型的现有信息，然后加载保存文件的内容。例如，如果保存文件是使用 SSdbsave -m 选项创建的，并只包含模型（而不是建模目录），则 SSdbload -l 选项不影响现有的建模目录。但是，它会删除任何现有模型，并将这些模型替换为保存文件中包含的模型。

-models

从您使用 <SAVE_FILE> 变量指定的保存文件中加载模型。

-catalog

从指定的保存文件中加载建模目录（模型类型、关系和规则）。

-replace

仅在容错环境中用于将当前分配到特定 SpectroSERVER 的优先级值替换为您使用 <PRECEDENCE> 变量指定的新值。

-add

仅在容错环境中用于通过 <PRECEDENCE> 变量将优先级值分配到特定的 SpectroSERVER。

-port

与 -add 或 -replace 参数结合使用以指定 SpectroSERVER 的端口号。如果不使用 <PORT_NO> 变量指定特定端口，将按默认使用 .vnmrc 文件中为 comm.port 资源指定的端口。

-showmap

输出格局映射信息，以显示在哪些服务器上以哪种优先级加载了哪些格局。

-version

显示 SSdbload 的版本，以及您尝试加载的已保存文件的版本。如果这些版本不兼容，将显示错误消息（即使您在静默模式下操作），并且不加载文件。

-extension

禁用强制文件扩展名。

-new_primary

当您加载保存在其他格局中的数据库时，必须使用此参数。使用 <NEWHOSTNAME> 变量以及从中加载数据库的主机的名称来为格局命名。否则，将为格局指定它最初保存到的 SpectroSERVER 中的主机名。

<SAVE_FILE>

要加载的备份文件的名称。备份可以是使用 SSdbsave 实用工具或联机备份功能保存的文件。也可以是 CA Spectrum 安装程序创建的保存文件之一。

每次成功执行此程序后，都会在 \$SPECROOT/SS 目录中创建两个保存文件，每个文件包含所安装的建模目录的副本。第一个文件是带有日期戳的文件，其扩展名为 .after。然后，将创建名为 “legacy.SSdb” 的 .after 文件副本，并覆盖之前存在的任何 legacy.SSdb 文件。

使用 legacy.SSdb 文件和最近安装的建模目录来重新初始化数据库。根据需要 使用 .after 文件还原与特定安装关联的目录。

.after 文件名中的日期戳后面的顺序计数器可让您区别在同一日期创建的多个文件，例如：

```
db_20001014,1.after.sldb  
db_20001014,2.after.SSdb  
db_20001014,3.after.SSdb
```

如果用于初始化数据库的 .after 文件早于当前可执行文件，则预期的信息可能不会出现在数据库中，并且不可通过 OneClick 访问。

示例

要初始化数据库并从之前保存的名为 “db_950318_cm” 的文件加载建模目录（模型类型和关系）与模型信息，您可以使用以下命令：

```
SSdbload -il db_950318_cm
```

错误

无法打开数据库。

SpectroSERVER 数据库不在当前目录中。

数据库已被锁定: <用户 ID>

进程: <进程名称>

进程 ID: <进程 ID>

位于网络节点: <节点名称>

启动时间: <日期/时间戳>

数据库由其他进程锁定。

保存文件版本: <version_number>。

SSdbload 版本: <version_number>。该版本的 SSdbload 无法加载保存文件。

CA Spectrum 版本 <version_number> 无法加载此保存文件。它将保存为版本 <version_number>。

如果检测到版本不兼容, SSdbload 将不允许您加载保存文件。如果您指定了 [-version] 参数, 则会生成此错误消息。即使您指定了 -quiet 选项, 在适用的情况下也会显示此消息。

CA Spectrum 版本 <version_number> 无法加载此保存文件。它将保存为版本 <version_number>。

如果检测到版本不兼容, SSdbload 将不允许您加载保存文件。如果您指定了 [-version] 参数, 则会生成此错误消息。即使您指定了 -quiet 选项, 在适用的情况下也会显示此消息。

注意: 如果在加载期间检测到非致命的属性描述符错误, SSdbload 将在包含数据库的目录中创建名为 SSdbload.log 的日志文件。以下是此文件中的非致命属性条目的示例:

重要说明! <SSdbload 路径>/SSdbload 无法读取模型 400000 中的属性 10004

详细信息:

[删除数据库锁定 \(p. 20\)](#)

更改主机名

如果更改单个 SpectroSERVER 主机的主机名, 您不必在进行更改之前先使用 SSdbsave, 然后再使用 SSdbload。数据库将自动更改以反映新的主机名, 当您重新启动 SpectroSERVER 时, 控制面板中会显示以下消息:

此数据库以前加载在 <旧主机名> 端口 <旧端口号> 上, 但是现在正在 <新主机名> 端口 <新端口号> 上加载。

但是，在包括一个或多个备份 SpectroSERVER 的容错环境中，服务器将通过一个与其主机名关联的 *优先级值* 来识别它们彼此之间的关系。因此，要保持容错关系，请按以下顺序使用 **SSdbsave** 和 **SSdbload** 来更改主要（或辅助）SpectroSERVER 的主机名：

1. 使用带 **-cm** 选项的 **SSdbsave** 保存数据库。
2. 更改主机名。
3. 通过使用 **-il** 和 **-replace** 选项运行 **SSdbload**，重新加载包含步骤 1 中创建的保存文件的数据库。通过重新加载，数据库可将新的主机名与现有优先级值进行关联：

```
SSdbload -il -replace <precedence> <save file>
```

详细信息：

[SSdbsave](#) (p. 44)

[SSdbload](#) (p. 40)

SSdbsave

此实用工具程序位于 `$SPECROOT/SS-Tools` 目录中，可让您创建现有 SpectroSERVER 数据库的建模目录（模型类型和关系）和/或其包含的实际模型与关联数据的备份副本。

在执行 **SSdbsave** 之前，必须关闭 SpectroSERVER 以及访问 SpectroSERVER 数据库的任何其他程序。

该实用工具采用以下格式：

```
SSdbsave [-quiet] [-extension] [-version]  
[-catalog] [-models] <SAVE_FILE>
```

其中，参数名称的第一个字母以粗体显示，您只需使用该字母，而不必输入整个名称。

-quiet

禁用交互/详细模式。

-extension

禁用实施文件扩展名。

-version

显示 **SSdbsave** 的版本。保存文件中包含了版本号。如果在保存文件时使用此参数，将会显示一条消息，指明可用于加载已保存文件的 **SSdbload** 版本。

-models

在保存文件中包括模型。

-catalog

在保存文件中包括建模目录（模型类型、关系和规则）。

<SAVE_FILE>

指定已保存数据库的目标文件的名称。保存文件的建议命名方案是包含日期戳以及用于保存的选项标志。例如，名为 `db_20121014_cm` 的文件表示在 2012 年 10 月 14 日同时使用 “c”（目录）和 “m”（模型）选项执行了备份。

示例输出

以下是使用 `-models` 和 `-catalogs` 参数运行 `SSdbsave` 时生成的输出的示例 (`../SS-Tools/SSdbsave -mc Mar16_2013DB`)。要保存的数据库文件的名称为 `Mar16_2013DB`。此输出的最后一行指示可用于加载这个已保存文件的 `SSdbload` 版本号。

保存的模型类型数: 3493

保存的关系数: 92

保存的模型数: 103

`SSdbsave` 已成功地将数据库模型和目录信息保存为 “`Mar16_2013DB.SSdb`”。

可以使用 `SSdbload` 版本 7.0.0.000 加载此文件。

错误**无法打开数据库**

SpectroSERVER 数据库不在当前目录中。

数据库已被锁定: <用户 ID>

进程: <进程名称>

进程 ID: <进程 ID>

位于网络节点: <节点名称>

启动时间: <日期/时间戳>

数据库由其他进程锁定。

SSdbsave: 警告。找不到所需的属性 0x999999, 模型: 0x999999, 模型类型: 0x999999 Cs 任何模型类型。处理将会继续

当尝试保存没有相应属性描述符的模型属性值时, `SSdbsave` 将显示此消息。如果用户或开发者删除了属性, 就会出现此信息消息。

数据库模型转换工具 (DBconv)

此实用工具程序位于 `$SPECROOT/SS-Tools` 目录中，可让您将 CA Spectrum 数据库中的一组模型从一个模型类型转换为另一个模型类型。DBconv 还可用于重新发现应用程序并在一组模型中重新配置接口。

该实用工具采用以下格式：

```
DBconv [-file=]<输入文件名>  
[-src_mth=]<源模型类型句柄>  
[-dest_mth=]<目标模型类型句柄>  
[-landscape=]<格局句柄>  
[-rediscover=]原因: <d><i>  
[-all_landscapes] [-test=]<测试级别> [-quiet] [-debug]
```

DBconv 和配置

DBconv 可以从输入文件（在命令行上指定）或命令行本身接收其配置。要获取命令行选项的列表，请运行不带选项的 DBconv。

-file

指定要使用的输入文件。

-src_mth

指定要转换的源模型类型句柄。如果指定了 `-rediscover`，请指定重新发现的模型类型。

-dest_mth

指定转换到的目标模型类型句柄。在输入文件中覆盖设置。如果指定了 `-rediscover`，将忽略此选项。

-landscape

指定要在其中搜索模型的格局句柄。

默认值: 0x400000

-all_landscapes

指定要在所有格局中搜索模型。

-test

测试级别 0（默认值）表示真正发生了转换。测试级别 1 表示 DBconv 在验证命令行和输入文件后立即停止处理。测试级别 2 表示未删除旧模型，并且未创建新模型。

-quiet

指定除错误消息以外，不应提供输出。

-debug

指定不应提供附加输出。

-rediscover

如果指定此选项，则不转换找到的模型。而是对找到的每个模型执行下列一个或多个指定操作：

d

销毁找到的每个设备模型的所有应用程序模型。

r

向找到的每个设备模型发送“重新发现应用程序”操作。

i

向找到的每个设备模型发送“重新配置接口”操作。

注意：可以在命令行上按任意顺序指定上述操作。但是，始终会按此处显示的顺序执行这些操作。

DBconv 命令行用法示例

以下是 DBconv 工具在命令行上的典型用法示例：

- 只使用指定的输入文件运行转换：

```
$ DBconv -file=config.dbc
```

- 在所有已知的格局中，将所有 GnSNMPDev 模型转换为智能交换机路由器：

```
$ DBconv -src_mth=0x3d0002 -dest_mth=0x2c60000 -all_landscapes
```

- 在格局 0x400000 中，将所有 GnSNMPDev 模型转换为智能交换机路由器：

```
$ DBconv -src_mth=0x3d0002 -dest_mth=0x2c60000 -landscape=0x400000
```

- 对于每个格局中的每个智能交换机路由器，销毁所有应用程序，重新发现应用程序，然后重新配置接口：

```
$ DBconv -src_mth=0x2c60000 -rediscover=rdi -all_landscapes
```

- 对于格局 0x400000 中的每个智能交换机路由器，重新配置接口：

```
$ DBconv -src_mth=0x2c60000 -rediscover=i -landscape=0x400000
```

注意：运行时，DBconv 将经历一个较短的初始阶段以及一个较长的第二阶段，并在窗口中指明生效的阶段。无法联系的模型将不转换；对于每种情况，都会显示错误消息。这些错误消息的某些可能原因为：

- 由指定模型类型代表的设备已失去联系
- 数据库中不存在指定的模型类型

- SpectroSERVER 未响应
- SpectroSERVER 中没有对应于您用户 ID 的模型

将 DBconv 与输入文件结合使用

可以结合或不结合输入文件使用 DBconv。如果要使用输入文件，请使用 `<${SPECROOT}>/SS-Tools` 目录中的 `template.dbc` 文件作为起点。

输入文件由多个部分组成，这些部分可让您以远高于命令行参数的精度自定义转换。输入文件由多个部分组成，除了 **Configuration** 部分外，其他所有部分都是可选的。请根据下面的指定使用部分格式和大小写。将忽略空白行和以“#”字符（英镑符号）开头的行。

DBconv 输入文件的元素

以下部分介绍了要结合 DBconv 使用的输入文件的必需和可选内容。

配置（必需）

此部分包含单个配置项，如下所述。

注意：在输入文件的“配置”部分中，只有 `Source_Model_Type`、`Destination_Model_Type` 和 `Landscape_Handle` 字段是必需的。其他所有字段都是可选字段。

示例：

```
Configuration {
Source_Model_Type = 0x3d0002
Destination_Model_Type = 0x2c60000
Reconnect_Sleep_Time = 90
Dont_Change_Discovery_Attributes = true
Models_To_Convert = 0
Landscape_Handle = all
Is_Obsolete_Model_Type = false
Relation_Section_Ignores = false
}
```

- `Source_Model_Type = <旧模型类型句柄>`

要转换的源模型类型句柄。它必须是十六进制数，并使用“0x”作为前缀。

- `Destination_Model_Type = <新模型类型句柄>`

转换到的目标模型类型句柄。它必须是十六进制数，并使用“0x”作为前缀。

- `Landscape_Handle = <格局句柄>`

指定要在其中完成转换的格局。如果指定了“all”，则 DBconv 将切换到企业模式，并转换所有格局中的模型类型。如果指定了“selection”，则将使用 `Landscapes { }` 部分中指定的所有格局。

- **Models_To_Convert = <要转换的最大模型数>**

如果为此行指定的值大于零，数据库转换操作只会转换指定数目的旧类型模型。如果值为 0，将转换所有模型。
- **Reconnect_Sleep_Time = <休眠时间>**

允许 **Model_State** 更改为活动的时间与尝试端口重新连接的时间之间存在可配置的休眠期（以秒为单位）。默认值为 60（秒）。
- **Reconnect_Interval = <秒>**

运行 **Reconnect_Sleep_Time** 之后，DBconv 将尝试重新连接接口和端口。如果模型在此时间后仍未变为活动状态（某些设备模型只有在创建了所有模块和应用程序之后才变为活动状态），则 DBconv 在重试之前将等待指定的时间。默认值为 30（秒）。
- **Reconnect_Interval_Count = <计数>**

此选项控制 DBConv 尝试重新连接接口和端口的次数。默认值为 30（次）。
- **Dont_Change_Discovery_Attributes = <true/false>**

如果值为 **false**，转换程序将修改旧模型类型中的发现相关属性。这会强制发现使用新模型类型而不是旧模型类型。在以下情况下，此参数默认为 **true**: (a) 旧模型类型为 **GnSNMPDev**，或 (b) 旧模型类型与新模型类型相同。默认值为 **false**。
- **Is_Obsolete_Model_Type = <true/false>**

此开关用于转换模型类型中设置了 **Obsolete** 标志的模型（如果设置此标志，您将无法读取任何属性）。在此情况下，转换期间不会读取/验证/写入任何属性。不会访问任何接口或端口。但是，将处理模型的关系。这意味着，如果转换带有此标志的设备模型，则创建的新模型将不带名称、IP 地址等。默认值为 **false**。
- **Relation_Section_Ignores = <true/false>**

此开关用于解释 **Left_Relationships** 和 **Right_Relationships** 部分。如果值为 **false**，则只会将 **Left/Right** 部分中指定的关联还原到设备模型（如果 **Left/Right** 部分均为空，则不还原任何关联）。如果值为 **true**，则会还原除指定关联以外的其他所有关联。这不影响设备模型左侧的 **CONNECTS_TO**，因为 DBconv 始终会尝试还原此项，以维护设备建模所在的视图。

- **Convert_Scm_Configs = <true/false>**

此开关用于确定是否连同模型类型一起转换 SCM 配置。如果此参数设置为 **true**，则会同时转换代表要转换的模型/模型类型的 SCM 配置。共享的 SCM 配置将转换为目标模型类型，非共享的 SCM 配置将转换为目标模型句柄和模型类型。

注意：将有效地传输主机 SCM 配置，但属性配置在转换期间可能会丢失。

- **Convert_Sanm_Policies = <true/false>**

此开关用于确定是否连同模型类型一起转换 SANM 策略。如果此参数设置为 **true**，则对任何策略中的源模型类型名称的任何引用将转换为目标模型类型名称。

- **格局**

要转换的格局列表。为了使 DBconv 检测到此部分，“Configuration”部分中的 Landscape_Handle 条目应设置为“all”。

示例：

```
Landscapes {  
0x400000  
0x80c00000  
}
```

- **Model_Handles**

要转换的模型列表。如果您只想要转换选定的模型，请在此部分中输入其模型句柄。将忽略所有其他模型。

示例：

```
Model_Handles {  
0x80c00be6  
0x80c00be8  
}
```

- **Transfer_Attributes**

要从旧模型传输到新模型的属性列表。确保这些属性对于旧模型类型和新模型类型均有效。请注意，由于始终会传输网络地址和团体名称属性，因此，无需在此部分中指定这些属性。

示例：

```
Transfer_Attributes {  
0x001006e  
0x00010024  
}
```

- **Transpose_Attributes**

要从旧模型转置到新模型的属性列表。确保这些属性对于旧模型类型和新模型类型均有效。旧属性和新属性的类型*必须*相同。

示例:

```
Transpose_Attributes {  
0x001006e = 0x001884  
0x00010024 = 0x777533  
}
```

- **Set_Attributes**

数据强制到的属性的列表。请验证这些属性是否有效，并且数据是否适用于属性的类型。

示例:

```
Set_Attributes {  
0x001006e = Router 42  
0x00010024 = public  
}
```

- **Transfer_Port_Attributes**

要从旧模型端口保存到新模型端口的属性列表。

示例:

```
Transfer_Port_Attributes {  
0x00011564  
}
```

- **Right_Relationships**

关系列表，这些关系与右侧旧模型之间的关联已被忽略或包含。（有关详细信息，请参阅本部分中的 **Relation_Section_Ignores** 开关。）

示例:

```
Right_Relationships {  
0x10004  
0x230000  
}
```

- **Left_Relationships**

关系列表，这些关系与左侧旧模型之间的关联已被忽略或包含。（有关详细信息，请参阅本部分中的 **Relation_Section_Ignores** 开关。）

示例:

```
Left_Relationships {  
0x10004  
0x230000  
}
```

- **Object_ID_Exists**

必须在每个目标设备上存在的 MIB 对象的列表。如果所有 MIB 对象存在并可读，则可以转换目标设备。

示例：

```
Object_ID_Exists {
1.3.6.1.4.1.52.2501.1.270.4.1.1.5
1.3.6.1.4.1.52.2501.1.1.5
}
```

- **Filter_Attributes**

用于选择模型的属性列表。如果发现任一旧模型中的 `attribute_id` 的 `attribute_value` 等于 `Filter_Attributes` 行中指定的值，则会选择模型。对于 `text_string` 属性，`attribute_value` 只需是从旧模型返回的字符串的子字符串。

示例：

```
Filter_Attributes {
0x10053 = 1.3.6.1.4.1.49.2.3.5
0x10b5a = 网络管理员
0x1154f = false
}
```

- **Advanced_Transfer_Attributes**

启用从关联到设备模型（通过已知关系路径）的任意数目的模型传输属性。可以在输入文件中包含其中任意数目的部分。

语法：

```
Advanced_Transfer_Attributes {
Path = <关系路径>
Identifier = <标识符属性>
If <属性 ID> = <值>
If ModelType = <模型类型 ID>
Transfer = <要传输的属性>
Transpose <目标属性 ID> = <源属性 ID>
}
```

说明：

Path = <关系路径> [强制]

指定如何查找要传输的模型。关系路径指定了一组以“.”分隔的关系名称。关系路径中的元素数目没有限制。

示例：

Path = HASPART

查找设备模型的 HASPART 关系右侧的所有模型，即所有接口。

Path = Contains.HASPART

扫描 Contains 关系右侧的所有模型。对于其中的每个模型，将扫描 HASPART 关系右侧的模型，找到的这些模型就是要对其执行传输的模型。

- Identifier = <标识符属性> [强制]

当找到了关系路径指定的模型集时，需要使用一种方法将这些模型与已转换数据库中的模型进行匹配，因为模型句柄将发生更改。DBconv 将读取“标识符”指定的属性的值。然后，尝试将此值与转换的模型进行匹配。如果找到了匹配项，DBconv 会将此模型视为等效于预先转换的模型。此条目的通用属性为 Component_OID。

示例：

Identifier = 0x1006a

If <属性 ID> = <值>

这是一个筛选条目。仅当 <属性 ID> 的值等于 <值> 时，它才会传输或转置指定的属性。

示例：

仅当 ifType 等于 6 时，以下三行才会匹配 Component_OID 标识的所有接口：

Path = HASPART

Identifier = 0x1006a

If 0x1134c = 6

If ModelType = <模型类型 ID>

仅当模型类型与 <模型类型 ID> 匹配时，才会传输或转置指定属性的筛选条目。

示例：

以下条目将匹配 RS-8000 上的所有 SSR_PortIf 接口。

Path = HASPART

Identifier = 0x1006a

If ModelType = 0x2c60006

- **Transfer = <要传输的属性>**

指定要传输的属性。条目数没有限制。需要一个 **Transfer** 条目或一个 **Transpose** 条目。

示例:

```
Transfer = 0x11564
```

- **Transpose <目标属性 ID> = <源属性 ID>**

指定要转置的属性。它采用 <源属性 ID> 中指定的值，并将该值写入 <目标属性 ID>。这两个属性的类型应该相同。需要一个 **Transfer** 条目或一个 **Transpose** 条目。

示例:

以下条目将复制 RS-8000 上的 VLAN 名称，并将该名称写入“注释”字段。

```
Transpose 0x11564 = 0x2c604dd
```

以下条目将从 RS-8000 上的所有 SS-PortIf 模型复制注释和轮询状态:

```
Advanced_Transfer_Attributes {  
  Path = HASPART  
  Identifier = 0x1006a  
  If ModelType = 0x2c60006  
  Transfer = 0x11564  
  Transfer = 0x1154f  
}
```

- **Transfer_Notes**

启用从指定区域复制“注释”字段。每行都将转换为 **Advanced_Transfer_Attributes**（已在本节中的前面介绍）。

示例:

```
Transfer_Notes {  
  设备  
  接口  
  应用程序  
  模块  
  端口  
}
```

可以指定一个或多个字段来复制指定类（设备、接口等）中的所有模型的“注释”字段。

第 3 章： DDM 数据库维护

此部分包含以下主题：

[数据库安全性](#) (p. 55)

[数据库大小管理](#) (p. 56)

[数据库备份](#) (p. 56)

[数据库还原](#) (p. 59)

[数据库维护和优化](#) (p. 60)

[DDM 数据库查询](#) (p. 62)

[其他数据库实用工具](#) (p. 63)

[Archive Manager 资源](#) (p. 64)

[DDM 和 Archive Manager 故障排除](#) (p. 65)

数据库安全性

要增强分布式数据管理器数据库的安全性，请在安装 CA Spectrum 后执行下列平台特定的过程之一。

Windows 系统中：

1. 以 Administrator 帐户登录。
2. 使用 Windows 资源管理器导航到 $\$SPECROOT/SS/DDM$ 目录。
3. 右键单击 .configrc 文件并选择“属性”。
4. 单击“安全”选项卡，检查“权限”面板，然后单击“高级”按钮。
5. 确保只列出了“Administrator”或“CA Spectrum Users”组。

如果您看到“Everyone”或者任何其他个人或组，请将其删除。可能需要清除允许将父级的可继承权限传播到子对象的设置。此设置在权限的“高级”选项下。

在 Solaris 上：

1. 以 root 用户身份登录。
2. 导航到 $\$SPECROOT/SS/DDM$ 目录。
3. 输入以下命令：

```
chmod 600 .configrc
```

数据库大小管理

管理 DDM 数据库的大小有助于提高它的运行效率。要控制 DDM 数据库的大小，您可以限制对网络关键组件的事件和统计信息日志记录。要管理数据库大小，您可以执行以下步骤：

- 通过禁用统计信息的日志记录来限制记录的数据。将 .vnmrc 文件中的 stats_logging_enabled 参数设置为 FALSE。

注意：

- stats_logging_enabled 的默认值为 FALSE。但是，如果您已从上一版本升级，请验证此设置。
- 有关 .vnmrc 文件的详细信息，请参阅《分布式 SpectroSERVER 管理员指南》。

- 限制存储的事件数据量。在 EventDisp 文件中针对其历史记录不重要的任何事件禁用“E”日志记录标志。

注意：有关详细信息，请参阅《事件配置用户指南》。

- 针对您未主动监控的模型，禁用日志记录或降低日志记录频率。您可以针对单个模型禁用日志记录，方法是在“模型信息”视图将“日志比率”（每个日志的轮询数）设置为零。或者，使用属性编辑器来针对特定类型的所有模型禁用日志记录。例如，如果您对用户端口的通信统计信息报告不感兴趣，则可以在所有用户端口上将“轮询日志比率”设置为零。

要降低日志记录频率，请使用这些方法之一来增大日志比率。

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》。

数据库备份

与 SpectroSERVER 数据库一样，DDM 数据库需要定期备份，以便在发生硬件或操作系统故障时能够完全恢复。在理想的情况下，这两个数据库应每日备份，以确保当前数据可用并保持数据库同步。

但是，DDM 数据库通常比 SpectroSERVER 数据库大得多，因此备份需要更多的时间。此外，在备份期间必须关闭 Archive Manager。因此，在重新启动 Archive Manager 时，经常需要处理大量积压的历史数据。积压数据处理可能会给 SpectroSERVER 的性能带来负面影响。出于这些原因，我们通常建议每周备份 DDM 数据库一次，并以更高的频率（最好是每天）备份 SpectroSERVER 数据库一次。

注意: 如果您只是还原了 DDM 数据库和 SpectroSERVER 数据库中的一个, 则两个数据库可能会不同步。例如, 您在星期日备份了两个数据库。在星期一创建了模型, 然后遇到系统崩溃, 则可使用星期日保存的文件还原 SS 数据库。还原的 SS 数据库不再包括您创建的模型, 但是该模型的历史记录仍在 DDM 数据库中。此外, 如果您重新创建了模型, 则随后您无法将原始模型中的 DDM 记录与针对重新创建的模型生成的新记录进行核对。为了避免这种冲突, 每当您还原 SpectroSERVER 数据库时, 都应该还原 DDM 数据库。

您随时可以使用 [ddm_save](#) (p. 58) 或 [ddm_backup](#) (p. 58) 工具执行 DDM 数据库备份。但是, 您也可以使用联机备份功能自动备份 DDM 数据库。联机备份使用 `post_olb_script`。因为保持 SpectroSERVER 和 DDM 数据库的同步非常关键, 因此, 该脚本是 DDM 备份的首选方法。

重要说明! 本节中介绍的工具 (`post_olb_script`、`ddm_backup` 和 `ddm_save`) 是唯一受支持的 DDM 数据库备份方法。使用第三方备份软件可能会导致数据库损坏。

post_olb_script

此脚本文件可让您协调执行自定义脚本以及使用联机备份执行 SpectroSERVER 数据库备份。该脚本位于 `$SPECROOT/SG-Support/CsScript` 目录中。当您使用默认设置运行它时, 在指定日期 (默认为星期日) 完成任何联机备份后, `post_olb_script` 将立即备份 DDM 数据库。

要使用默认设置运行 `post_olb_script` 文件, 请将该文件从 `$SPECROOT/SG-Support/CsScript` 复制到 `$SPECROOT/custom/CsScript`。使用文本编辑器删除每行开头的英镑符号 (`#`), 从而激活该文件的最后 11 行。DDM 数据库备份排定在星期日 (目标日期=0) 执行的任何联机备份之后。因此, 如果您将联机备份排定为每日运行, 则 DDM 数据库将每周自动备份一次。

注意: 在 CA Spectrum 升级安装期间, 将自动执行复制操作。在复制之前, 请确认该文件尚不存在, 以避免恢复为默认设置。

如果您需要提高 DDM 数据库备份频率, 请仅注释掉 `post_olb_script` 文件的最后五行。这样便不会指定“目标日期”。因此, 每次使用联机备份备份 SpectroSERVER 数据库之后, 就会备份 DDM 数据库。

详细信息:

[联机备份](#) (p. 11)

ddm_save

ddm_save 实用工具位于 `$SPECROOT/SS/DDM` 目录中，用于将整个历史数据库保存到指定的文件中。在执行该命令之前，请使用控制面板关闭 Archive Manager，然后在完成保存时重新启动它。

注意： ddm_save 实用工具不会从 DDM 数据库中删除任何数据；它会将数据复制到指定的文件。

该实用工具采用以下格式：

```
ddm_save [-extension] [-quiet] <SAVE_FILE>
```

其中，参数名称的第一个字母以粗体显示，您只需使用该字母，而不必输入整个名称。

-extension

默认情况下，ddm_save 将为文件分配 .tgz 文件扩展名。如果指定此参数，保存的文件将不带文件扩展名。

-quiet

禁用交互/详细模式。

<SAVE_FILE>

指定已保存数据库的目标文件的名称。

注意： ddm_save 命令会将必要的数据库文件保存为默认文件扩展名为 .tgz 的 gzip tar 文件格式。

ddm_backup

ddm_backup 实用工具位于 `$SPECROOT/SS/DDM` 目录中，用于关闭 Archive Manager，执行 ddm_save 命令，以及在完成保存时重新启动 Archive Manager。

该实用工具采用以下格式：

```
ddm_backup <SAVE_FILE>
```

<SAVE_FILE>

指定已保存数据库的目标文件的名称。提供包括目标文件名的完全限定文件路径以便在该位置保存文件。如果未指定完全限定的文件路径，将在 DDM 目录中保存该文件。

详细信息：

[ddm_save](#) (p. 58)

有关备份的要点

- 备份的频率越高，丢失数据的风险就越低。
- 备份历史数据不会从 DDM 数据库中删除这些数据。
- 将根据 `.configrc` 文件中的 `MAX_STAT_DAYS` 和 `MAX_EVENT_DAYS` 设置从 DDM 数据库中清除（永久删除）历史记录。有关详细信息，请参阅 [Archive Manager 资源](#) (p. 64)。
默认设置为 45 天。一旦任何记录达到指定的期限，随后就会每天发生清除。因此，每天都会清除最早的数据。
- 如果使用 `ddm_save` 方法，则在完成备份后，必须使用“控制面板”重新启动 Archive Manager。

数据库还原

在发生硬件或操作系统故障时，请通过加载使用任意受支持备份方法（`post_olb_script`、`ddm_backup` 或 `ddm_save`）创建的保存文件来还原 DDM 数据库。

可以使用下一节将要介绍的 `ddm_load` 工具加载保存文件。

`ddm_load`

此实用工具位于 `$SPECROOT/SS/DDM` 目录中，可让您还原 DDM 数据库。首先，它将初始化 DDM 目录（从该目录中删除所有数据），然后加载您指定的保存文件。如果您在加载保存文件或目录，则会自动发生初始化。仅当您想要清除 DDM 目录并从头开始时，才需要使用 `-initialize` 选项。如果使用该选项，则还需要指定关联 SpectroSERVER 的格局句柄。

在执行该命令之前，请通过控制面板关闭 Archive Manager。然后，在完成还原后，重新启动 Archive Manager。

该命令采用以下格式：

```
ddm_load [-quiet] [-initialize <LANDSCAPE_HANDLE>] [-events_init]
[-stats_init][<SAVE_FILE>]
```

其中，参数名称的第一个字母以粗体显示，您只需使用该字母，而不必输入整个字符串。

-quiet

禁用提示（交互模式）。用于从脚本内部运行加载命令。

-initialize

从 DDM 数据库中删除所有数据。只能结合 LANDSCAPE_HANDLE 参数使用。

<LANDSCAPE_HANDLE>

指定正在初始化其数据库的 SpectroSERVER 的格局句柄。

-events_init

仅初始化（删除）DDM 数据库中的事件记录。

-stats_init

仅初始化（删除）DDM 数据库中的统计记录。

<SAVE_FILE>

指定要加载的备份文件的名称。ddm_load 实用工具既支持新的 gzip tar 文件格式，也支持传统的专有文件格式。

有关还原的要点

- 要保持数据库同步，每次还原 SpectroSERVER 数据库之后，必须还原 DDM 数据库。
- 当您重新加载给定保存文件中的某些记录时，这些记录有可能达到了其清除日期。因此，在重新启动 Archive Manager 后，将立即清除这些记录。
- 完成还原时重新启动 Archive Manager。

数据库维护和优化

以下各节介绍您可以运行哪些脚本来保持 DDM 数据库的高效运行。

db_maintenance.pl

此脚本位于 `$SPECROOT/SS/DDM/scripts` 目录中，可从 DDM 数据库中删除所有未引用的统计记录。在删除模型时，通常会创建未引用的记录。随着时间的推移，它们的关联记录将被清除，在数据库中留下一些无用的残余数据。

重要说明！ 只能在已关闭 **Archive Manager** 的情况下运行 **db_maintenance.pl** 脚本。请仔细权衡运行此脚本与关闭 **Archive Manager** 的利弊。

请不要将此脚本排定为频繁运行。您可以偶尔运行此脚本，或者在从格局中删除了极大量的模型时运行此脚本。运行此脚本实现的性能提升并不明显。如果销毁了大量的模型（如整个格局），请只针对常规内务管理运行此脚本。

建议在运行此脚本之前运行 **db_optimize.pl** 脚本。此外，建议在对生产环境影响最小的时候运行此脚本。针对未经优化的大型数据库或包含许多事件记录的数据库运行此脚本可能需要很长时间。

此脚本采用以下格式：

```
./db_maintenance.pl [-q]
```

-q

（静默）禁用提示（交互模式）。

详细信息：

[db_optimize.pl](#) (p. 61)

db_optimize.pl

此脚本位于 `$SPECROOT/SS/DDM/scripts` 目录中。它提供一种简便的方法，供您优化 **DDM** 数据库中的所有表。

优化表可以带来两个主要好处：

- 提高发送到 **DDM** 数据库的查询的速度。
- 回收清除记录后产生的磁盘空间。

您可以根据性能要求，将此脚本排定为每周或每月运行。请记住，此脚本可能需要相当长的时间才能完成运行，具体取决于 **DDM** 数据库分段的大小和级别。

注意： **db_optimize.pl** 脚本所需的可用磁盘空间超过了 `$SPECROOT/mysql/data/ddmdb` 目录中最大文件的大小。建议在运行此脚本之前备份您的数据库文件。

可以在启动 **Archive Manager** 后运行该脚本。但是，它将在每两次表优化之间产生 10 分钟延迟，使 **Archive Manager** 能够从锁定的数据库中恢复。

此脚本采用以下格式：

```
./db_optimize.pl[-q]
```

-q

（静默）禁用提示（交互模式）。

DDM 数据库查询

本节中提供的查询可帮助您确定正在生成哪些事件，以及哪些模型正在生成这些事件。我们已提供查询来帮助您确定特定的设备是否正在生成过多的事件，或者是否需要对统计数据的存储天数进行更改。根据 DDM 数据库事件表的大小，这些查询可能需要一段时间才能完成运行。

注意：有关设置 MAX_STAT_DAYS 参数值（控制数据的存储天数），请参阅 [Archive Manager 资源](#) (p. 64)。

登录 MySQL:

在命令提示符下，从 `$SPECROOT/mysql/bin` 中输入以下命令：

```
mysql -uroot -proot ddmb
```

在 UNIX 中，还传递以下参数：

```
-S $SPECROOT/mysql/tmp/mysql.sock
```

此时将显示 `mysql>` 提示符。

要查询 DDM (ddmb) 数据库，请执行以下操作：

在 `mysql>` 提示符下，输入下列任一查询：

- 显示表统计信息：

```
SHOW TABLE STATUS LIKE "event";
```

- 获取事件的总计数：

```
SELECT COUNT(*) FROM event;
```

- 获取在特定日期后发生的事件计数：

```
SELECT COUNT(*)  
FROM event  
WHERE utime >= UNIX_TIMESTAMP("yyyy-mm-dd");
```

- 获取最经常生成的前 10 个事件：

```
SELECT HEX(type), COUNT(*) AS cnt
FROM event
GROUP BY type
ORDER BY cnt DESC
LIMIT 10;
```

- 获取发生的事件数最多的前 10 个模型：

```
SELECT HEX(e.model_h), m.model_name, COUNT(*) AS cnt
FROM event e, model m
WHERE e.model_h=m.model_h
GROUP BY e.model_h
ORDER BY cnt DESC
LIMIT 10;
```

- 获取发生的事件数最多的前 10 个日期：

```
SELECT DATE(FROM_UNIXTIME(UTIME)) AS x, COUNT(*) AS cnt
FROM event
GROUP BY x
ORDER BY cnt DESC
LIMIT 10;
```

- 获取过去 10 天的事件量：

```
SELECT date(from_unixtime(utime)) AS x, COUNT(*) AS cnt
FROM event
GROUP BY x
ORDER BY x DESC
LIMIT 10;
```

其他数据库实用工具

CA Spectrum 提供三个 MySQL 数据库实用工具，它们都安装在 `$SPECROOT/mysql/bin` 目录中。

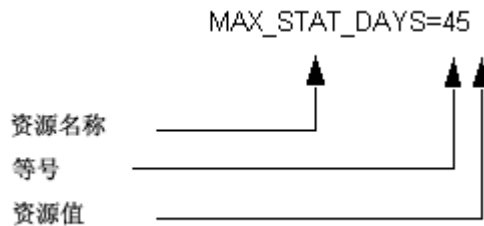
- 当 MySQL 服务器停止时，您可以使用 `myisamchk` 检查和修复 MyISAM 表。
- `mysqlcheck` 类似于 `myisamchk`。但是，它不要求您停止服务器来检查或修复表。此实用工具还能优化和分析表。
- `mysqladmin` 可让您执行管理操作，例如检查服务器配置或丢弃数据库。

注意：有关使用这些 MySQL 实用工具的详细信息，请参阅 <http://www.mysql.com> 中提供的文档。

Archive Manager 资源

Archive Manager 资源是控制历史记录处理方式的设置。通过这些资源，Archive Manager 还能够与 SpectroSERVER 和 DDM 数据库进行通信。

Archive Manager 资源在 .configrc 文件中定义，该文件位于 $\$SPECROOT/SS/DDM$ 目录中。资源以 “resource = resource_value” 的格式列出，如下所示。



.configrc 文件定义以下 Archive Manager 资源：

ARCH_MGR_SOCKET_NUMBER

标识 Archive Manager 在其上侦听来自 VNM 和 SSAPI 客户端的请求的端口。

默认值： 0xbafe

AUTO_REPAIR_DB

控制 Archive Manager 是否自动尝试修复损坏的 DDM 数据库。默认情况下，此参数未在 .configrc 文件中列出，因此，不会启用自动修复。要启用自动修复功能，请将 AUTO_REPAIR_DB=TRUE 添加到 .configrc 文件。要禁用自动修复功能，请将值设置为 FALSE，或从 .configrc 文件中删除该条目。

默认值： 已禁用

DDM_DATABASE_NAME

Archive Manager 与 DDM 数据库之间的通信需要此参数。安装 CA Spectrum 时已自动读入默认值。不建议更改此值。

默认值： ddmdb

DDM_DATABASE_PORT

指定用于连接 MySQL 数据库服务器的端口。

DDM_DATABASE_HOSTNAME

Archive Manager 与 DDM 数据库之间的通信需要此参数。安装 CA Spectrum 时将自动设置默认值。不建议更改此值。

默认值： 本地主机

DDM_DATABASE_PASSWORD

Archive Manager 与 DDM 数据库之间的通信需要此参数。安装期间将自动读入默认值。不建议更改此值。

默认值: CA Spectrum 安装所有者用户名密码

DDM_DATABASE_USERNAME

Archive Manager 与 DDM 数据库之间的通信需要此参数。安装期间将自动读入默认值。不建议更改此值。

默认值: CA Spectrum 安装所有者用户名

DDM_SOCKET_NUMBER

已弃用。此资源不会在将来的版本中出现。

LANDSCAPE_PRECEDENCE

当前不支持的资源。

MAX_DB_CONNECTIONS

指定 Archive Manager 为请求提供服务时可以使用的最多并发 MySQL 连接数。

默认值: 25

MAX_EVENT_DAYS

指定事件数据的最大存储天数。当超出了 MAX_EVENT_DAYS 时，将清除较旧的数据。

默认值: 45

MAX_STAT_DAYS

指定统计数据的最大存储天数。当超出了 MAX_STAT_DAYS 时，将清除较旧的数据。

默认值: 45

TIME_TOLERANCE_IN_SECONDS

指定为记录数据报告的时间戳与实际日志记录时间之间的最大允许差异（以秒为单位）。时间容差值越大，压缩数据的效率就越高，从而可以节省 CPU 和磁盘资源。但是，增大此值会使准确度降低。

默认值: 300

DDM 和 Archive Manager 故障排除

本节阐述与 Archive Manager 和 DDM 数据库相关的问题的可能原因与解决方案。

ARCHMGR.OUT 中的错误消息

症状

会看到以下消息：

<date/time> ArchMgr.cc(591) 上出错：未能打开与 <主机名> 上的 SpectroSERVER 的连接，0xbeef。该错误表示 SpectroSERVER 已关闭。将在 60 秒后重试。

解决方案

SpectroSERVER 未准备就绪。Archive Manager 在启动 SpectroSERVER 时会自动启动，但只有在 SpectroSERVER 准备就绪后才能连接。在首次启动或者每当 SpectroSERVER 关闭时总会显示此消息。

症状

在连接或重新连接 Archive Manager 后，将显示以下消息：

<date/time> : ArchMgr 已成功注册到 DDM 名称服务 - <ip 地址> (<计算机名称>)。

症状

在尝试启动 Archive Manager 时，会看到以下消息：

<date/time> : ArchMgr 正在关闭...

解决方案

由于 SpectroSERVER 不包含尝试启动它的用户的 CA Spectrum 用户模型，Archive Manager 已关闭。

症状

会看到以下消息：

<date/time> : ArchMgr 已成功关闭。

如果此消息持续出现，请查看 OneClick 控制台中的“用户”选项卡，以确认 ArchMgr 进程的所有者是否存在对应的用户。

症状

会看到以下消息：

```
<date/time> : ArchMgr 已经以用户 “<user_name>” 的身份启动
```

```
<date/time> : ArchMgr 正在验证数据库
```

检测到数据库损坏：

```
    ddmdb.statistic_ul64 - 记录 delete-link-chain 已损坏
```

```
    ddmdb.statistic_ul64 - 损坏
```

打开 DDM 数据库时出错。一个或多个表已损坏。

恢复选项：

1. 运行 “ArchMgr -repair” 以尝试恢复
2. 使用 “ddm_load <savefile>” 加载有效的 DDM 保存文件
3. 使用 “ddm_load -i <LANDSCAPE_HANDLE>” 初始化 DDM 数据库

```
<date/time> : ArchMgr 数据库无效错误
```

解决方案

DDM 数据库已损坏，无法加载。您可以使用 `ddm_load` 命令来加载保存的 DDM 数据库或初始化 DDM 数据库。也可以尝试使用 `ArchMgr -repair` 命令修复现有的数据库文件。

要运行 `ArchMgr -repair` 命令，请导航到 `$SPECROOT/SS/DDM` 目录并输入以下内容：

```
./ArchMgr -repair
```

成功完成修复后，将显示以下消息：

“ArchMgr 已成功修复数据库”。

要使 Archive Manager 自动尝试修复损坏的 DDM 数据库，请在 `.configrc` 文件中使用 `auto_repair_db` 选项。有关详细信息，请参阅 [Archive Manager 资源](#) (p. 64)。

未列出事件

症状

“事件”选项卡中未列出任何事件，并且“服务器(连接状态)”按钮以黄色闪烁。当您双击该按钮时，“连接状态”对话框为事件和视图服务显示“关闭”状态，但为格局服务显示“启动”状态。

解决方案

Archive Manager 未运行或未连接到 SpectroSERVER。查看“控制面板”中的“控制”菜单以确认 Archive Manager 服务是否已启动。如果已启动，请检查 `$SPECROOT/SS/DDM/ARCHMGR.OUT` 文件以确定是否存在 Archive Manager 错误。

