CA Spectrum®

Modeling Gateway Toolkit Guide（建模网关工具包指南）

版本 9.4

CA Technologies 产品引用

本文档引用以下 CA Technologies 产品：

- CA Spectrum® (CA Spectrum)
- CA Spectrum® 建模网关工具包（建模网关）
- CA CMDB

联系技术支持

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

目录

第 1 章： 建模网关概述

此部分包含以下主题：

建模网关先决条件

使用 CA Spectrum Modeling Gateway 工具包之前，请确保您：

- 比较全面地了解 CA Spectrum。
- 阅读《*概念指南*》。
- 具有 XML 的应用知识。
- 理解文档类型定义 (DTD) 的概念。
- 详细了解要导入的网络拓扑。
- 可以使用 UNIX 或 Windows 在文件系统中导航、复制和删除文件，以及创建和编辑文本文件。

关于建模网关工具包

通过 CA Spectrum 建模网关工具包，集成人员可以在 CA Spectrum 中导入和导出网络拓扑数据。该工具包包括一个定义 XML 元素和属性的文档类型定义 (DTD)，还包括一个定义 CA Spectrum 语法以及要导入或导出的信息的资源文件。

要使用 DTD 元素进行拓扑导入，您可以创建一个描述网络上的设备、端口和连接的 XML 文件。此 XML 文件可以在 CA Spectrum 中创建新的拓扑数据、更新现有数据或销毁不再正确的数据。此外，在 XML 语法中使用的元素和属性可以进行扩展和自定义，以满足大部分集成的需要。

通过此工具包，您还可以使用逗号分隔的 ASCII 文本文件导入帧中继或 ATM 连接。您也可以使用上述 XML 功能导入该连接信息。

一旦 CA Spectrum 中存在网络拓扑数据，您就可以像管理其他手动创建或通过发现创建的模型一样来管理这些设备。您可以查看导入结果，以及有关每次导入的任何诊断信息。

通过建模网关工具包，您还可以使用 XML 文件从 CA Spectrum 导出拓扑信息和配置设置。此信息随后可通过建模网关导入到指定的 SpectroSERVER。

给 CA Spectrum 装备关于正在进行的基础的动态会聚网络拓扑结构信息先前是艰难的任务。发现和手动建模并不适用于千变万化的环境所必需的持续更新。使用发现的建模连接对于各种物理基础架构也是一个挑战，例如下列环境中的物理基础架构：

- 有线 MSO（多业务运营商）
- ATM（异步传输模式）
- 帧中继

CA Spectrum 建模网关可有效地解决这些问题。

导入体系结构

要进行导入，需要在导入集成过程中从第三方数据库提取数据，并创建一个输入文件。此输入文件可能是一个 XML 文件或一个逗号分隔的 ASCII 文件，具体取决于内容。XML 输入文件可提供最大范围的导入选项，是本指南的重点。通过逗号分隔的文件，您可以为帧中继和 ATM 电路创建连接。

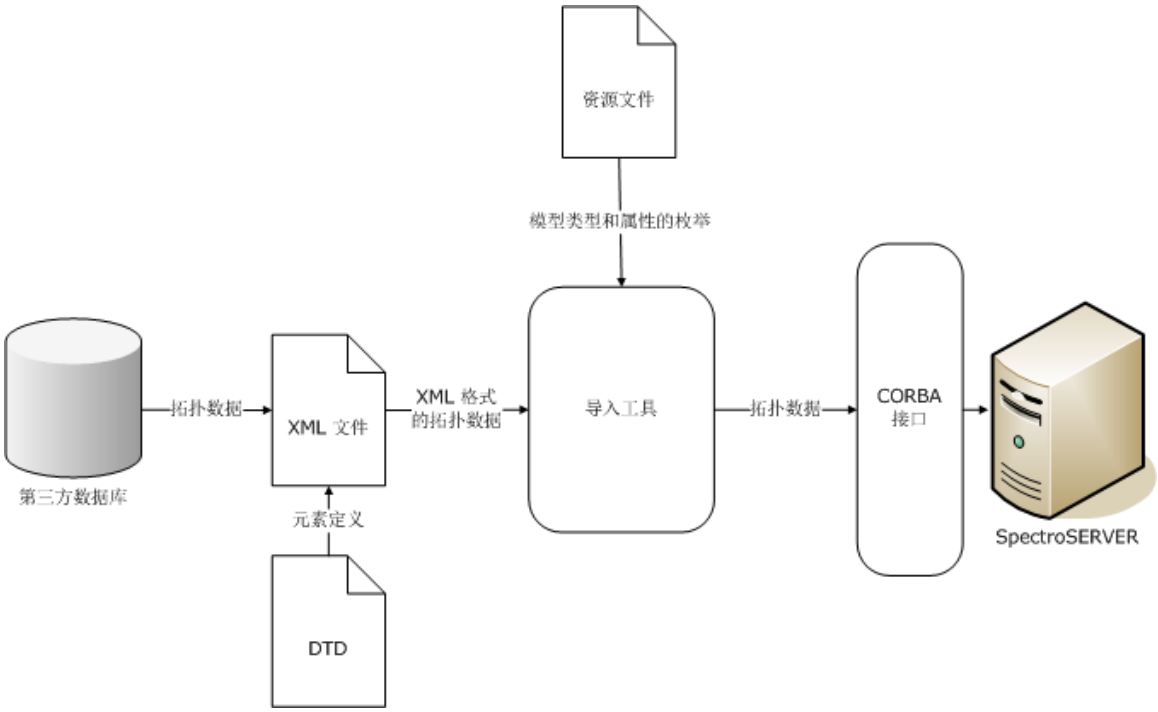
在创建 XML 输入文件时，请使用所提供的文档类型定义 (DTD) 和 .modelinggatewayresource.xml 文件。DTD 定义 XML 元素、属性及其关联的语法规则。.modelinggatewayresource.xml 文件显示可供使用的 CA Spectrum 模型类型和属性。此文件将 CA Spectrum 模型类型名称和属性名称与 CA Spectrum 用于该模型类型或属性的唯一十六进制标识符联系起来。.modelinggatewayresource.xml 文件可以自定义，以满足您的特定集成的需要。

创建第一个输入文件后，它可作为表示同一类型输入的多个数据集的模板。例如，如果您创建了一个用于导入设备的 XML 文件，则可以通过替换特定于设备的拓扑数据来重复使用该文件。

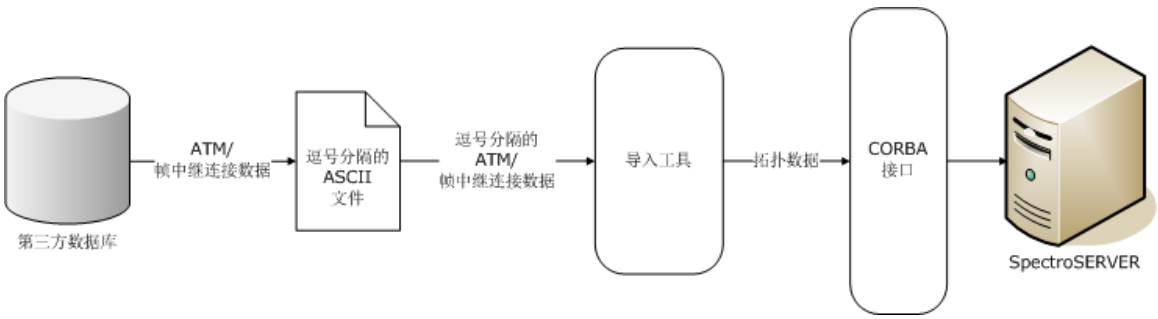
modelinggateway 工具是一个命令行实用工具，可用于从输入文件读取网络拓扑信息并将数据发送至 SpectroSERVER 数据库。利用 XML 文件的数据，导入工具可以创建、销毁和更新连接、设备和容器模型。它还可用于从 CA Spectrum 导出数据。

CA Spectrum 建模网关还提供了多种机制，来验证每次数据库导入的安全性和准确性。例如，它可以维护审核跟踪记录，其中包含与每一次的创建、删除、关联和更新操作相关的记录。您可以查看有关 OneClick 中导入的信息。如果在导入过程中发生任何类型的关键故障，则 CA Spectrum 会生成事件以报告错误。所有错误及其可能的原因都记录在错误日志文件中。您也可以启用调试日志，以帮助您查找问题或不准确之处的根源。

下图显示了建模网关如何使用 XML 文件导入数据。数据从第三方数据库流入，并通过 DTD 和 .modelinggatewayresource.xml 文件在 XML 文件中格式化，以符合语法。导入工具随后会对 XML 文件进行解释，并通过 CA Spectrum CORBA 接口将数据发送至 CA Spectrum。



下图显示了建模网关如何使用逗号分隔的 ASCII 文本文件导入帧中继和 ATM 连接信息。



导出体系结构

要进行导出，将 SpectroSERVER 作为资源，建模网关可以将拓扑数据和配置数据导出到 XML 文件。此 XML 文件随后可以与第三方工具集成，或重新导入到另一个 SpectroSERVER。例如，可以重新导入此文件，以进行 CA Spectrum 分区或格局句柄更改。

第 2 章： 将拓扑数据导入 CA Spectrum

此部分包含以下主题：

如何使用建模网关进行导入

您可以使用 Modeling Gateway 将第三方拓扑数据导入 CA Spectrum。首先，从第三方数据库提取拓扑数据，并将其在输入文件中格式化。

要使用 Modeling Gateway 将第三方拓扑数据导入 CA Spectrum，请执行以下任务：

1. 提取拓扑数据。

   从第三方数据库提取网络拓扑数据。由于各数据库系统有所不同，请参阅您所使用的数据库的相关文档完成此步骤。

2. 在输入文件中格式化数据 (p. 12)。

   创建一个 XML 文件或一个逗号分隔的输入文件，以便为导入工具格式化数据。

3. （可选）移动 modelinggateway 工具。

   注意：要在另一台服务器上运行 modelinggateway 工具，请将 modelinggateway 工具及其所有支持文件移至该服务器。有关详细信息，请参阅 *《分布式 SpectroSERVER 管理员指南》*。

4. 运行 modelinggateway 工具 (p. 28)。

   创建输入文件后，使用导入工具将数据发送至 CA Spectrum。

5. （可选）导入逗号分隔的文件 (p. 29)。

   您可以在不使用导入工具的情况下从 OneClick 界面导入帧中继和 ATM 连接数据。

6. 查看导入信息 (p. 30)

   在 OneClick 中验证导入的进度和结果。

注意：不要使用 Modeling Gateway 将模型从一个版本的 CA Spectrum 迁移至另一版本。对实体进行标识和建模所用的方法在不同的 CA Spectrum 版本之间可能不同。因此，不要使用 Modeling Gateway 导入从不同版本的 CA Spectrum 导出的 XML 文件。

在输入文件中格式化数据

使用建模网关导入数据时，可以使用两种类型的输入文件：XML 文件和逗号分隔的文件。XML 输入文件可用于创建或销毁模型和连接，以及更新属性值和连接信息。随建模网关一起提供的 DTD 中的语法定义了要在 XML 输入文件中使用的元素。逗号分隔的文件只能用于创建 ATM 和帧中继连接。以下几部分概述了如何创建每种类型的输入文件。

XML 输入文件

要了解用于创建 XML 输入文件的元素，需要熟悉 CA Spectrum 对网络基础架构建模所用的过程。以下部分概要说明了这一过程，并讨论了它如何应用于在 XML 输入文件中使用的 XML 元素。如果您了解这些概念，则可以跳过此部分，并转到 XML 输入文件语法 (p. 15)。

详细信息：

XML 输入文件语法 (p. 15)

分层视图

CA Spectrum 视图是一种数据组织方式，便于数据的显示或操作。分层视图表示您的网络数据的构建方式。在 XML 文件中构建网络数据时，请从表示各分层视图的元素中进行选择。分层视图有两种：拓扑视图和位置视图。

拓扑视图

拓扑视图实际上是网络组件的抽象化。使用拓扑视图时，您表示的是您网络中的物理或逻辑组件，对这些组件进行分组时应考虑其逻辑连接。您还可以选择使用显示端口级别或设备级别设备连接情况的管道，以图形方式来表示连接。在 OneClick 中，此视图显示为 Universe 拓扑。

注意：有关 OneClick 中的可用拓扑视图的详细信息，请参阅《*IT 基础架构建模与管理 - 管理员指南*》。

位置视图

位置视图按照物理位置来组织您的网络数据。使用位置视图时，可以通过地理措词来描述您的网络。您可以从全球办事处开始。然后，直接转至您办事处所在各个地区各建筑物各楼层的配线柜。在 OneClick 中，此视图显示为 World 拓扑。

注意：有关 OneClick 中的可用拓扑视图的详细信息，请参阅《*IT 基础架构建模与管理 - 管理员指南*》。

模型和模型类型

在 CA Spectrum 中预定义了许多模型类型。当模型类型在 CA Spectrum 界面中实例化来表示特定网络实体时，它们称为模型。模型类型主要分两大类：

- 智能模型类型
- 容器模型类型

智能模型类型可实例化来表示在网络上运行的实际设备。它们有 IP 地址和 MAC 地址，且 CA Spectrum 可使用 SNMP 直接与这些设备进行通信。容器模型类型实例化后变成主要用于模型分组的模型。

模型可根据所用分层视图的类型进行分组。例如，您可以使用拓扑视图创建 LAN 模型，以便对您某段网络上的特定设备进行分组。或者，您可以使用位置视图创建机房模型，以便将您的建筑物中一个机房内的所有设备归为一组。

一个容器模型可包含其他容器模型、智能模型或两者，具体取决于特定的模型类型。例如，一个网络容器模型可能包含一个智能模型来表示路由器。该网络容器模型还可能包含一个 LAN 容器模型来表示一个 IP 地址范围。另一方面，建筑物模型只能包含容器模型；例如，楼层、分区或机房。

通过 DTD 中的元素，您可以使用任何分层视图及其相应容器模型类型来描述您的网络拓扑。您也可以使用任何可实例化的智能模型类型。智能模型类型并非取决于所使用的分层视图的类型。

注意：如果需要，您可以指定模型句柄而不是模型类型来标识模型。在指定模型句柄时，建模网关将忽略其他任何模型标识符，仅使用模型句柄来标识模型。

实际上，并不是所有在 CA Spectrum 知识库中定义的模型类型都可用于在 OneClick 中创建模型。一些模型类型用作派生出其他模型类型的基础模型类型。

注意：有关详细信息，请参阅《*概念指南*》。

建模方法

在 CA Spectrum 中对设备建模的方法有两种。第一种方法是使用设备的 IP 地址或 DNS 名称。利用此信息，CA Spectrum 联系设备，并使用最适合表示设备功能的模型类型来创建模型。

第二种方法是提供模型类型来创建设备模型。您仍须提供 IP 地址或 DNS 名称，以便 CA Spectrum 可以与设备进行通信。但是，无论 CA Spectrum 对设备功能的评估如何，您选择的模型类型或模型句柄都将实例化。

要创建容器模型等非设备模型，必须提供模型类型和模型名称以便导入。

CA Spectrum 属性

每个模型类型都有一组关联属性。每个属性都在某些方面对模型类型进行了描述。实例化模型的属性所具有的值反映了该模型所表示的设备并描述了该模型当前的状态。例如，模型类型 Host_Sun 具有属性 IPAddress。如果 Host_Sun 模型类型的模型已实例化，则此属性的值反映了该模型所表示的设备的 IP 地址。

XML 语法还使用术语属性。XML 属性描述有关元素的更多信息。在 CA Spectrum 建模网关的 XML 语法中，一些 XML 属性用于赋值给 CA Spectrum 属性。

注意：CA Spectrum 所定义的属性称为 CA Spectrum 属性；常规 XML 属性简称为属性。

XML 输入文件语法

生成 XML 文件时，请使用 DTD 文件中定义的语法规则。以下部分概要说明了各 DTD 元素的功能。

下列说明和示例并未涵盖各元素的所有属性。有关各元素及其属性的完整参考，请参阅文档类型定义元素 (p. 39)。

根元素

在 DTD 中定义的元素存在于与 CA Spectrum 中的网络表示类似的层次结构中。必须与每个 XML 导入文件一起使用的根元素是导入元素。XML 语法规则指定根元素是最外层的元素，且表示 XML 文件的开始和结束。因此，您文档中所用的其余 XML 元素由导入元素所围绕。

面向模型的元素

面向模型的元素定义您网络中的物理或逻辑组件。它们是容器类型元素，用于创建定义网络元素分组逻辑方法的模型。根据元素所在 CA Spectrum 分层视图的类型进行分组。每个此类容器类型元素都可存在于特定分层视图中的一个分层视图中。

- Topology_Container
- Location_Container
- Device
- Schedule
- Port
- Connection
- GenericView_Container

Topology_Container

Topology_Container 元素可创建根据物理或逻辑拓扑对其他模型进行分组的模型。Topology_Container 元素可创建容器模型，因此应使用 model_type 属性或模型句柄来标识您要使用的特定容器。可能的 model_type 值在 DTD 中枚举。LAN 是一个 Topology_Container model_type 值的示例。请使用 name 属性来指定 Topology_Container 的名称。name 和 model_type 属性唯一地标识所创建的模型。如果您指定了模型句柄，则 name 和 model_type 属性将被忽略。Topology_Containers 可包含其他 Topology_Container 元素、设备或连接。Topology_Container 模型始终置于 OneClick 拓扑视图中。

注意：默认情况下，name 和 model_type 属性赋值给 CA Spectrum 属性 Model_Name 和 Modeltype_Name。但是，您可以通过编辑 .modelinggatewayresource.xml 文件来更改 name 属性赋值的目标 CA Spectrum 属性。无论选择了什么新 CA Spectrum 属性（与模型类型一起），都将用于唯一标识该容器。这种更改允许两个容器具有相同的模型名称。

Location_Container

Location_Container 元素根据物理或地理位置对其他模型进行分组。建筑物和机房都是 Location_Container 元素模型类型值的示例。Location_Container 元素可创建容器模型，因此应使用 model_type 属性或模型句柄来标识您要使用的特定容器。可能的 model_type 值在 DTD 中枚举。请使用 name 属性来指定 Location_Container 的名称。name 和 model_type 属性唯一地标识所创建的模型。如果您指定了模型句柄，则 name 和 model_type 属性将被忽略。

注意：默认情况下，name 和 model_type 属性赋值给 CA Spectrum 属性 Model_Name 和 Modeltype_Name。但是，您可以通过编辑 .modelinggatewayresource.xml 文件来更改 name 属性赋值的目标 CA Spectrum 属性。无论选择了什么新 CA Spectrum 属性（与模型类型一起），都将用于唯一标识该容器。这种更改允许两个容器具有相同的模型名称。

Device

Device 元素定义网络上的设备。此元素与其他元素一起使用，可创建、更新或销毁 CA Spectrum 中设备模型的实例。使用 SNMP 设备时，请提供有效、唯一的 IP 地址或 DNS 名称来唯一标识使用 ip_dnsname 属性的设备。通过此唯一标识，CA Spectrum 可与该设备进行通信，并根据设备功能选择最适合的模型类型。应将 ip_dnsname 设置为有效字符串。如果 ip_dnsname 无效或无法连接，则设备模型创建可能失败。如果提供给 model_type 的 ip_dnsname 无效或无法连接，则仍使用指定的模型类型创建设备模型。但是，设备模型未激活，不能提供任何有效的网络信息或状态。设备可能的 model_type 值在 .modelinggatewayresource.xml 文件中枚举。

Schedule

Schedule 元素定义设备模型置于维护模式的时间。当设备模型处于维护模式时，设备及其组件的管理通信将被挂起。挂起通信可防止在设备维护期间 CA Spectrum 生成有关设备模型的任何事件或警报。

Port

当您创建设备模型时，会自动创建端口。通过 Port 元素，您可以修改一些 CA Spectrum 端口属性值或指定端口级别连接。您可以指定不同类型的端口，包括帧中继或 ATM 电路。要标识设备上的端口，请提供 identifier_name 和 identifier_value 属性的值。可能的 identifier_name 值在 DTD 中枚举。identifier_value 是 identifier_name 属性选择的标识符的值。Port 元素必须始终指定为 Device 元素的子项。

Connection

Connection 元素定义两台设备之间的物理或逻辑连接，包括 WAN 链路连接，因此必须包含两个 Device 子元素。如果在 Device 元素中指定了 Port 元素，则连接将在该设备的指定端口上解析。如果 Device 元素没有指定 Port 元素，则由 CA Spectrum 发现尝试确定进行连接解析的端口。

GenericView_Container

要在常规视图中创建容器模型，请使用 GenericView_Container 元素。GenericView 元素和 GenericView_Container 元素都可用于创建自定义视图。因此，您作为集成人员，需要决定什么情况下使用此容器以及如何使用。

面向任务的元素

在 DTD 中定义的其余元素是面向任务的元素。这些元素及其属性可帮助定义输入文件所生成的操作的类型。通过这些元素，您可以创建新的拓扑信息，更新、覆盖或删除现有的拓扑信息。单个输入文件可使用每个此类元素中的零个或一个，但 Connection 元素除外。您可以根据需要使用多个 Connection 元素。

- Topology
- Location
- GenericView
- Connection
- Update
- Destroy

新的拓扑数据

面向任务的元素定义您要对 XML 文件采取的操作。当您要在 CA Spectrum 中创建新的网络拓扑数据时，请使用 Topology 元素和 Location 元素。这些元素定义您要在哪个分层视图中创建数据。然后，您可以使用相应的模型元素作为子元素，来为网络实体创建模型。要自定义视图以满足您的特定集成需求，请使用 GenericView 元素。

创建位置视图

您可以在位置视图中创建您的拓扑信息，以便在 OneClick 的 World 拓扑中查看。要在位置视图中创建此信息，请在 Import 根元素内使用 Location 元素构建 XML 文件。Location 元素可包含用于创建特定容器模型的 Location_Container 元素或用于创建设备模型的 Device 元素。

示例：位置视图中的站点容器

以下示例在位置视图中创建一个站点容器，并在该容器中创建一个设备。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
        <Location>
                <Location_Container  model_type = "Site" name = "My_Town" >
                        <Device ip_dnsname= "10.253.9.18"
                                community_string="public"/>
                </Location_Container>
        </Location>
</Import>
```

Import 元素是根元素，且始终包含在输入文件内。

Location 元素指示您将在位置视图中创建模型。

Location_Container 元素创建容器模型。此模型是网络的逻辑组件而不是物理组件。因此，CA Spectrum 不能与其联系，也不能用 IP 地址或 DNS 名称定义模型类型。要指示即将创建的容器模型的类型，请提供 model_type 属性和 name 属性的值。可能的 model_type 属性值在 DTD 和以下部分 Location_Container (p. 52) 中列出。name 属性是必需的，而且必须为模型指定唯一名称。

注意：当指定模型句柄时，模型句柄用于标识容器模型。因此，如果指定了模型句柄，则 name 和 model_type 属性将被忽略。

Device 元素在站点 Location_Container 模型内创建模型。ip_dnsname 属性是 Device 元素的必需属性。如果可以联系设备，则 CA Spectrum 使用 IP 地址或 DNS 名称来查找设备。

有关这些元素及其可能属性的完整参考，请参阅文档类型定义元素 (p. 39)。

创建拓扑视图

您可以将您的网络数据导入拓扑视图中，以便在 OneClick 的 Universe 拓扑中查看。要将数据导入拓扑视图，请在 Import 根元素内使用 Topology 元素构建 XML 文件。Topology 元素可包含：

- 用于创建特定类型的容器模型的 Topology_Container 元素。
- 用于创建特定类型的设备模型的 Device 元素。
- 用于在两台设备之间创建连接的 Connection 元素。

通过使用 DTD 中所述的层次结构和语法规则，您可以准确地表达您网络中的物理和逻辑连接。

示例：拓扑视图中的 LAN 容器

此示例在拓扑视图中创建一个 LAN 容器，并在该容器中创建一个设备。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
        <Topology>
                <Topology_Container  model_type = "Lan"
                        name = "Sample_LAN" Security_String = "public"
                        subnet_address= "10.253.9.0" subnet_mask =
"255.255.255.0">
                        <Device ip_dnsname= "10.253.9.18"
                                community_string="public"/>
                </Topology_Container>
        </Topology>
</Import>
```

Import 元素是根元素，且始终包含在输入文件内。

Topology 元素指示您将在拓扑视图中创建模型。

Topology_Container 元素创建容器模型。此模型是网络的逻辑组件而不是物理组件。因此，CA Spectrum 不能与其联系，也不能用 IP 地址或 DNS 名称定义模型类型。要指示即将创建的容器模型的类型，请提供 model_type 属性和 name 属性的值。可能的 model_type 属性值在 DTD 和以下部分 Topology_Container (p. 72) 中列出。name 属性是必需的，而且必须为模型指定唯一名称。其他指定的属性是可选的。

注意：当指定模型句柄时，模型句柄用于标识容器模型。因此，如果指定了模型句柄，则 name 和 model_type 属性将被忽略。

Device 元素在 LAN Topology_Container 模型内创建模型。ip_dnsname 属性是 Device 元素的必需属性。如果 CA Spectrum 可联系设备，则使用 IP 地址或 DNS 名称查找设备。CA Spectrum 找到设备时，由它确定适合用来创建模型的模型类型。

在多个视图中表示同一设备

您可以创建一个 XML 文件，用于在多个视图中表示一个设备。要创建此文件，我们建议您用于创建此设备的模型的每个 Device 元素具有相同的属性和属性值。如果属性和属性值不同，那么导入工具将尝试合并这些 Device 元素的属性和值，以创建一组一致的属性和值。有时虽然为每个此类 Device 元素指定了属性，但使用的是不同的值。在这种情况下，列入 XML 文件的最后一个 Device 元素的值会覆盖之前用于创建该设备的模型的其他 Device 元素的所有属性值。

请记住，一些属性具有默认值。例如，community_string 属性的默认值为 public。因此，当您指定一个 Device 元素属性和值以表示设备 A 时，我们建议在其他所有表示设备 A 的 Device 元素中指定该属性和值。这样做有助于确保不会使用该属性的默认值覆盖先前指定的值。

示例：在拓扑视图和位置视图中创建设备

以下示例中，在拓扑视图和位置视图中创建了设备 10.253.9.16。您可以看到，这两个视图中用于描述设备的属性和值是一样的。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- 拓扑视图导出 -->
        <Topology discover_connections="false" complete_topology="false">
                <Device ip_dnsname="10.253.9.16" community_string="zippo" />
        </Topology>
<!-- 位置视图导出 -->
        <Location complete_topology="true">
                <Location_Container model_type="Site" name="Durham">
                        <Device ip_dnsname="10.253.9.16"
community_string="zippo"/>
                </Location_Container>
        </Location>
</Import>
```

使用发现来创建连接

CA Spectrum 连接表示两台设备之间的物理或逻辑链路。在 XML 输入文件中创建连接的方法有两种。第一种方法是借助 Topology 元素的 discover_connections 属性来使用 CA Spectrum 发现。当 discover_connections 属性设置为 true 时，会对新建的设备模型运行发现。随后，发现会为这些设备建立连接。

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

示例：使用发现来创建连接

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
        <Topology >
                <Topology_Container  model_type = "Lan" name = "Sample_LAN"
discover_connections= "true"
                        Security_String = "public" subnet_address= "10.253.9.0"
                        subnet_mask = "255.255.255.0" >
                        <Device ip_dnsname= "10.253.9.18"
                                community_string="public"/>
                        <Device ip_dnsname= "10.253.9.20"
                                community string="public"/>
                </Topology_Container>
        </Topology>
</Import>
```

使用 Connection 元素创建连接

创建连接的第二种方法是使用 Connection 元素，以便连接已创建的设备。连接可以在两个端口之间、一个设备与一个端口之间或两台设备之间指定。

通过指定设备之间的连接，CA Spectrum 可查明发生故障的设备，但首选指定端口级别连接。端口级别连接是更精细级别的连接，允许 CA Spectrum 在端口级别解析连接和分析故障。当您在两台设备之间指定了一个连接，而没有指定其中一个或两个用于连接的端口时，CA Spectrum 将自动尝试确定端口。如果此过程成功，则 CA Spectrum 将连接解析到端口级别。

以下两个条件同时出现时，CA Spectrum 将生成一个错误，指示连接失败：

- CA Spectrum 无法确定用于连接的两个端口。
- 至少有一个设备是可管理的设备。

错误将写入错误日志文件。

如果 CA Spectrum 只能确定其中一个端口，那么仅在连接的一侧将连接解析到端口级别。另一侧仍解析到设备级别。

如果两台设备都是不可管理的设备，CA Spectrum 将在设备级别建立连接。

示例：在现有端口之间创建连接

以下示例在两个分别属于不同设备的现有端口之间创建连接。Connection 元素标识要链接的 Port 和 Device 元素。连接将在两台设备的端口级别解析，因为在每个 Device 元素内都指定了一个 Port 元素。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
        <Connection>
                <Device ip_dnsname= "172.19.57.93">
                        <Port identifier_name = "frCircuitTableInstance"
                                identifier_value="4.161"/>
                </Device>
                <Device ip_dnsname = "192.168.125.161">
                        <Port identifier_name= "frCircuitTableInstance"
                                identifier_value= "2.861"/>
                </Device>
        </Connection>
</Import>
```

上一个示例在 DLCI 端口之间指定了连接。由于 identifier_name 属性的值是 frCircuitTableInstance，端口使用 MIB 中 frCircuitTable 对象的 OID 实例值来标识。OID 实例值使用 identifier_value 属性指定。

Connection 元素包含在 Topology 元素或 Topology_Container 元素中，以指示设备的分层位置。这种情况并不改变输入文件的结果。

重要说明！ 当您尝试导入的 XML 文件包含使用同一个 Port 元素的多个 Connection 元素时，Modeling Gateway 不会报告错误。单个端口不能有多个连接。如果在多个 Connection 元素中指定了同一个端口，XML 文件中的最后一个 Connection 元素将覆盖之前指定该端口的所有 Connection 元素。

详细信息：

错误日志 (p. 31)

创建 WA_Link 连接

要创建 WA_Link 连接，请使用以下语法：

```
<Connection>
    <Device ip_dnsname=10.253.9.18/>
    <Device ip_dnsmame=10.253.9.100 model_type="WA_Link">
</Connection>
```

建模网关会自动创建 WA_Segment。将在段和一台或多台设备之间创建链路。要在第二台设备和链路之间指定连接，请再向导入文件中添加一个连接标记。

在 CA Spectrum 和第三方数据库之间同步信息

Topology、Location、Topology_Container 和 Location_Container 元素都有一个名为 complete_topology 的属性。如果将此属性的值设置为 true，则指示 XML 文件定义 CA Spectrum 必须知道的所有模型和连接。当 XML 文件导入到 CA Spectrum 时，CA Spectrum 视图中不存在于 XML 文件内的所有模型将发送至“Lost and Found”。如果该视图中有子容器，CA Spectrum 将引用在指定此子容器的元素中设置的 complete_topology 属性的值。如果在子容器元素中未指定 complete_topology 属性值，则从父元素继承该值。因此，如果父元素的 complete_topology 设置为 true，且子容器元素没有指定 complete_topology 设置，那么子容器的 complete_topology 值也为 true。

当 CA Spectrum 导入 XML 文件时，如果出现以下条件，模型将发送至“Lost and Found”：

- 模型在正导入到的视图中或在该视图的子容器中。
- 模型*不*在 XML 文件中。

在将第三方数据库的数据与 CA Spectrum 的数据进行同步时，此行为很有用。

示例：Complete_Topology 设置为 true

以下示例中，在 Topology 元素中 complete_topology 设置为 true。拓扑视图中的所有现有模型将发送至“Lost and Found”，但在此输入文件中指定的模型除外。在此示例输入文件中，仅指定了两个模型：

- LAN Topology_Container
- IP 地址为 10.253.9.18 的设备

如果上述模型不存在，应创建这两个模型。如果上述模型存在，CA Spectrum 将使用输入文件中的属性值来更新其 CA Spectrum 属性值。拓扑视图（VNM 除外）存在的其他所有模型将发送至“Lost and Found”。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
        <Topology complete_topology="true">
                <Topology_Container  model_type = "Lan" name ="Sample_LAN"
                        Security_String = "public" subnet_address= "10.253.9.0"
                        subnet_mask = "255.255.255.0">
                        <Device ip_dnsname= "10.253.9.18"
                                community_string="public"/>
                </Topology_Container>
        </Topology>
</Import>
```

如果在 Topology_Container 元素而不是 Topology 元素中使用了 complete_topology 属性，那么 CA Spectrum 将沿着层次结构仅删除该 Topology_Container 中未指定的模型。

更新信息

要更新现有模型的 CA Spectrum 属性和关联信息，请使用 Update 元素。Update 元素可包含 Container 元素、Device 元素和 Association 元素。Port 属性的值使用相应的 Device 元素进行更新。

示例：更新两个不同模型的属性并创建关联

以下示例显示了一个更新输入文件。在此例中，两个不同模型的两个属性将更新，并在这两个模型之间创建关联。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
  <Update>

    <Topology_Container model_type="Lan" name="Sample"
                        model_name = "newLAN"/>
    <Device ip_dnsname= "Test1" poll_interval= "1108"/>

    <Association relation="0x10002">
       <Left_Model> <Topology_Container name="Net"
                                        model_type="Network" /></Left_Model>
       <Right_Model> <Device ip_dnsname="172.24.94.94" /></Right_Model>
    </Association>
  </Update>
</Import>
```

第一个更新的属性是 LAN 容器模型的 model_name 属性。模型名称从 Sample 更改为 newLAN。请注意，使用了下列属性：name 和 model_name。这两个属性之所以存在，是为了更改 CA Spectrum 属性 Model_Name。要标识容器模型，并使用 model_name 属性指定容器模型的新名称，请使用 name 属性并将当前名称作为值。

然后，此示例将设备轮询时间间隔值从 Test1 更改为 1108。将新值赋予 poll_interval 属性将覆盖旧值。

此示例还在模型类型为网络的容器模型“Net”和设备模型 172.24.94.94 之间创建了一个关系关联 0x10002，条件是这两个模型均存在于 SpectroSERVER 中。

销毁信息

使用 Destroy 元素可删除容器模型、设备模型、连接和关联。当您销毁设备时，所有与此设备关联的端口模型和应用程序模型也将被销毁。

示例：销毁 LAN 容器、连接和关联

以下示例中，将销毁名为 newLAN 的 LAN 拓扑容器。除非指定为销毁，否则此容器中的所有模型都将发送至“Lost and Found”。此示例还将销毁两台设备（即 Test1 和 Test2 ）之间在端口级别指定的连接。如果在模型类型为网络的容器模型“Net”和设备 172.24.94.94 之间存在“Collects”关联，也会予以销毁。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">

<Import>
    <Destroy>
        <Topology_Container model_type="Lan" name="newLAN"/>

        <Connection>
          <Device ip_dnsname= "Test1">
           <Port identifier_name= "ifIndex" identifier_value= "1"/>
          </Device>
          <Device ip_dnsname= "Test2">
           <Port identifier_name="ipAddress"
              identifier_value = "10.253.8.18"/>
          </Device>
        </Connection>

        <Association relation="Collects">
       <Left_Model> <Topology_Container name="Net"
                                        model_type="Network" /></Left_Model>
       <Right_Model> <Device ip_dnsname="172.24.94.94" /></Right_Model>
    </Association>

    </Destroy>

</Import>
```

重要说明！要销毁一个表示已从网络中删除的设备的模型，在指定 XML 文件的 Destroy 元素中的设备 ip_dnsname 属性时，请使用设备的 IP 地址而不是 DNS 名称。一旦设备已从网络中删除，该设备的 DNS 条目就不再存在。并且，建模网关将无法标识相应的要删除的模型。

.modelinggatewayresource.xml 文件

Topology_Container、Location_Container 和 Device 元素都有一个 model_type 属性，其值必须为有效的 CA Spectrum 模型类型。CA Spectrum 使用十六进制数唯一标识模型类型。这些十六进制值已在资源文件 .modelinggatewayresource.xml 中枚举。该文件将模型类型的文本值与唯一十六进制标识符组对。文本值随后显示在 DTD 中。

DTD 中定义的许多属性与 CA Spectrum 属性相对应。在 CA Spectrum 中使用十六进制数唯一标识 CA Spectrum 属性。.modelinggatewayresource.xml 文件不使用 DTD 或 XML 文件中的这些十六进制值。但是，该文件将 CA Spectrum 属性的十六进制标识符与更直观的基于文本的名称组对。

.modelinggatewayresource.xml 文件的 ModelType 元素和属性元素都可进行自定义。

注意：.modelinggatewayresource.xml 文件还用于从 CA Spectrum 导出拓扑数据。有关详细信息，请参阅从 CA Spectrum 导出拓扑数据 (p. 33)。

定义字符集编码

默认情况下，CA Spectrum XML 输入文件采用 UTF-8 进行编码。要导入特殊字符或外国语言，请在 XML 文件标头中指定相应的字符集编码，如以下示例所示。

示例：将输入文件字符编码设置为 Greek

以下示例显示了如何修改 XML 文件，以便将字符编码设置为 Greek：

```
<?xml version="1.0" encoding="ISO-8859-7" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
```

查看 CA Spectrum 字符集编码信息

如果需要确定 CA Spectrum 使用的字符集编码，您可以在 OneClick 管理页面完成此操作。

要查看 CA Spectrum 字符集编码信息

1. 在 OneClick 主页中单击“管理”。

   将打开“管理”页面。

2. 单击左侧面板中的“字符集”。

   此时将打开“字符集编码”页面，其中显示了编码及其相应语言的列表。

逗号分隔的输入文件

建模网关可在 XML 输入文件中指定 ATM 和帧中继连接。该工具包也可接受来自逗号分隔的 ASCII 文本文件的 ATM 和帧中继连接信息。此文件可用于导入有关以下电路间连接的信息：

- 两个 ATM 电路
- 两个帧中继电路
- 一个 ATM 电路和一个帧中继电路

您可以指定在 OneClick 中创建活动管道来表示连接。在同一个输入文件中可指定多个连接。

注意：这些连接中涉及的设备模型必须先存在于 CA Spectrum 中。

逗号分隔的输入文件的语法

以下示例显示了在逗号分隔的输入文件中使用的格式：

```
<Device_IP>, <OID>, <Device_IP>, <OID>, <CircuitName>, <CircuitID>, <Pipe>
```

Device_IP

指定连接中涉及的每个设备的 IP 地址。必需。

OID

指定 frCircuitTable、atmVclTable 或 atmVplTable 的 OID 实例，以指定设备的电路链路。

CircuitName

（可选）指定涉及的电路的名称。

CircuitID

（可选）指定涉及的电路的 ID。

Pipe

（可选）有两个可能的值：CREATE_PIPE 或 NO_CREATE_PIPE。如果值设置为 CREATE_PIPE，则将在指定的连接之间创建活动管道。如果值设置为 NO_CREATE_PIPE，则不会在指定的连接之间创建活动管道。如果没有为该参数指定值，则使用默认值 CREATE_PIPE。

示例：帧中继电路之间的指定连接

以下示例显示了指定两个帧中继电路之间的连接的输入文件。将在这两个端口之间创建一个活动管道。

```
172.19.57.93, 4.161, 192.168.125.161, 2.161, FR_Circuit_Name, Circuit_Id_123,
CREATE_PIPE
```

运行 modelinggateway 工具进行导入

要运行 modelinggateway 工具进行导入，请使用以下语法。

Windows

```
modelinggateway.bat -vnm vnm_name -i import_file [-o outputfile] [-debug
debugfile]
```

Solaris/Linux

```
modelinggateway -vnm vnm_name -i import_file [-o outputfile] [-debug debugfile]
```

-vnm *vnm_name*

指定 SpectroSERVER 主机的名称。

- i *import_file*

指定包含要用 .modelinggateway.dtd 编译的必要输入信息的 XML 文件的名称。

-o *outputfile*

（可选）将错误信息记录到在 *outputfile* 参数中指定的文件。如果未使用此选项，则错误信息将记录到名为 *import_file*.log 的文件。*Import_file* 是 XML 文件的名称。

-debug *debugfile*

（可选）指示您要在导入过程中创建调试输出文件。使用 -debug 选项时，您可以为输出提供自己的调试文件名称。如果不为 *debugfile* 提供值，则调试文件名称将默认为后缀为“.debug”的 *import_file* 名称。

注意：-debug 选项要求运行建模网关的计算机有磁盘空间。例如，如果 *import_file* 中的模型数量大，或设备模型的接口密度大，则可能会导致较大的调试输出文件。

ImportConfiguration 元素

要控制建模网关导入数据方式的特定方面，请使用 ImportConfiguration 元素。

ImportConfiguration 元素具有以下语法：

```
<ImportConfiguration
    do_not_process_pre_existing_devices_under_container_node = "false"
    import_to_primary_ss_only = "false"
    max_device_creation_threads = "50"
/>
```

do_not_process_pre_existing_devices_under_container_node

指定 CA Spectrum 是否处理在容器元素下找到的先前存在于 CA Spectrum 中的设备。

默认： False

import_to_primary_ss_only

指定当主要 SpectroSERVER 宕机时建模网关是否连接到备用 SpectroSERVER。

默认值： False

max_device_creation_threads

指定可以同时创建和激活多少设备模型。

注意： 如果将此值设置为高于 50，则可能导致过多的 SNMP 通信。

默认值： 50

导入逗号分隔的文件

前一部分介绍了如何使用 modelinggateway 导入工具导入输入文件。您也可以从 OneClick 界面导入帧中继和 ATM 连接数据。

遵循这些步骤:

1. 在 OneClick 控制台中单击适用的 VNM 模型。
2. 单击“组件详细信息”面板中的“信息”选项卡。
3. 单击“逻辑连接导入”展开这一部分。
4. 单击“导入”，找到包含您要导入到 CA Spectrum 的数据的逗号分隔文件，然后单击“打开”。

   OneClick 将导入数据。将显示“导入结果”对话框，为您提供有关导入成功的信息。

查看导入信息

CA Spectrum 建模网关提供了多种机制，以帮助确保每次数据库导入的安全性和准确性。CA Spectrum 建模网关可以维护审核跟踪记录，其中包含所执行的每一次创建、删除、关联和更新相关的记录。您可以查看有关 OneClick 中导入的数据，也可以跟踪错误日志和调试日志中有关导入问题的信息。

在 OneClick 中查看建模网关结果

依次单击 OneClick 控制台中的 VNM 模型、“信息”选项卡，可以验证建模网关导入结果。

遵循这些步骤:

1. 在 OneClick 控制台中单击适用的 VNM 模型。
2. 单击“组件详细信息”面板中的“信息”选项卡。
3. 单击“建模网关”以展开这一部分。
4. 查看“建模网关”部分中的表以了解有关最近导入操作的信息。该表包含以下信息：

   导入文件

   显示导入文件的名称。

   日志文件

   显示日志文件的名称。

   开始时间

   指示拓扑导入过程开始的时间。

   结束时间

   指示拓扑导入过程结束的时间。

   进度

   进度字段显示尚未完成的拓扑导入的状态。此字段可能的值有：

   - 正在初始化
   - 正在标识模型
   - 正在创建模型
   - 正在激活模型
   - 正在映射连接
   - 正在放置模型

- 正在创建连接
- 正在更新模型
- 正在销毁模型
- 完成
- 已断开连接

错误

显示在导入过程中生成的错误的数量。

模型已创建

显示在导入过程中创建的模型的数量。

模型已销毁

显示在导入过程中消除的模型的数量。

模型已更新

显示在导入过程中更新的模型的数量。

连接已创建

显示在导入过程中创建的连接的数量。

连接已删除

显示在导入过程中删除的连接的数量。

5. 单击“最大记录数”设置链接以修改表中列出的导入文件的数量。
6. 单击任何表列的标题以根据需要对数据进行排序。
7. 在“筛选”字段中输入文本，将导入数据限于特定标准。
8. 单击“更新”以检查导入随着处理进度的前进而变化的状态。

   屏幕将刷新并显示最新的可供使用的导入信息。

错误日志

所有错误及其可能的原因都记录在错误日志文件中。默认情况下，导入工具创建一个名为 *<nameofimportfile>*.log 的错误日志，其中 *<nameofimportfile>* 是您的导入文件的名称。您也可以使用在导入工具部分中指定的语法，为日志文件指定一个特定名称。当导入完成时，日志文件将显示在 SS-Tools 目录中。日志文件记录成功创建、删除和更新模型与连接的数量。日志文件还记录在导入过程中发生的每一次失败。

第 3 章： 从 CA Spectrum 导出拓扑数据

此部分包含以下主题：

关于从 CA Spectrum 导出拓扑数据

Modeling Gateway 支持从 SpectroSERVER 导出拓扑信息和配置设置。信息将导出到一个 XML 格式的文件中，随后可使用 Modeling Gateway 将此文件导入指定的 SpectroSERVER。

默认情况下，将导出以下类型的信息：

- 具有配置属性的 Device 元素。
- 具有配置属性的 Port 元素。
- 具有配置属性的 Container 元素。
- 连接（已解析的和未解析的，WA_Link 连接）。
- Universe 拓扑层次结构。
- Universe 拓扑中每种视图的布局，包括批注和缩放信息，但不包括背景图像。
- 用户模型和整个用户方案，如与用户相关的关系、属性以及 LicenseRole、AccessGroup、PrivilegeRole 和 UserGroup 等模型。
- 发现配置。
- 服务管理方案和属性。
- 静态和动态全局集合，包括每个全局集合中的所有模型、所有动态集合标准、已缩放列表、已分组列表以及拓扑布局。

重要说明！ converter 工具的目的是跨两个或多个 SpectroSERVER 对一个 SpectroSERVER 数据库进行分区，或更改单个 SpectroSERVER 数据库的格局句柄。如果您要使用 Modeling Gateway 导出功能来实现与 converter 工具实现的相同目的，请注意，并非所有信息都会导出。要为目前导出功能不支持的数据配置行为及设置建模，必须在将导出的数据导入后进行一些手动操作。不会被导出的数据类型示例包括，但不局限于，Service Performance Manager (SPM) 测试以及 Policy Manager 策略。

配置导出设置

您可以通过修改 .modelinggatewayresource.xml 文件中的 ExportConfiguration 元素来控制导出的数据。默认情况下，Universe 容器下的所有拓扑信息和建模信息都会被导出。您可以修改 RootContainerToExport 元素，以指定从中导出数据的不同根容器。根容器及其每个子容器的所有内容都会被导出。

注意：不需要使用 DTD 导出数据；DTD 仅用于导入数据。

为设备、容器和端口导出的属性分别在下列元素中定义：DeviceExportAttributes、ContainerExportAttributes 和 PortExportAttributes。根据需要从这些元素中添加和删除属性。

SpectrumConfigurationExport 元素控制当 ExportConfiguration 元素中的 export_spectrum_settings 标志设置为 true 时将导出哪种类型的 CA Spectrum 配置数据。

例如，以下元素控制为 LostFound 模型导出的属性：

```
<SpectrumConfigurationExport model_type="LostFound" >
     <Automatic_Model_Destruction attribute_id="0x11de1" />
     <Model_Destruction_Interval_Hours attribute_id="0x11de3" />
     <Model_Destruction_Interval_Minutes attribute_id="0x11de4" />
</SpectrumConfigurationExport>
```

详细信息：

ExportConfiguration 元素 (p. 34)

ExportConfiguration 元素

要控制导出行为，请使用 .modelinggatewayresource.xml 文件中的 ExportConfiguration 元素。

ExportConfiguration 元素具有以下语法：

```
<ExportConfiguration
    export_devices           = "true"
    export_containers        = "true"
    export_port_attributes   = "true"
    export_links             = "true"
    export_topology_layout   = "true"
    export_annotation        = "true"
```

```
    export_WA_Link_models    = "true"
    export_spectrum_settings = "true"
    export_user_models       = "true"
    export_service_modeling  = "true"
    export_schedules         = "true"
    export_global_collections = "true"
    export_discovery_configs = "true"
    export_from_primary_ss_only = "false"
    export_policy_manager = "true"
/>
```

export_devices

导出设备模型。

export_containers

导出容器模型。

export_port_attributes

导出端口属性。

export_links

导出设备链路。

export_topology_layout

导出设备模型和容器模型在拓扑中的 x, y 坐标。

export_annotation

导出批注和模型组信息。

export_WA_Link_models

导出 WA_Link 模型。如果您决定不导出 WA_Link 模型，则将其视为是透明的。两个设备模型之间的广域链路作为直接链路导出。

export_spectrum_settings

导出 CA Spectrum 设置，如故障隔离、发现和 VNM 控制的设置。

export_user_models

导出用户模型、用户许可、用户权限、用户首选项及所有其他与用户相关的关系、属性和模型。

export_servicemodeling

导出服务管理方案和属性。

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

export_schedules

导出排定。

export_global_collections

导出静态和动态全局集合，包括每个全局集合中的所有模型、所有动态集合标准、已缩放列表、已分组列表以及拓扑布局。

export_discovery_configs

导出发现配置。

export_from_primary_ss_only

指定当主要 SpectroSERVER 宕机时建模网关是否连接到备用 SpectroSERVER。

export_policy_manager

导出 Policy Manager 策略。所有相关模型、策略、规则、权限以及模板都包含在内。

示例：

以下示例将导出除端口属性信息之外的所有内容。此示例还告知建模网关当主要 SpectroSERVER 宕机时*不*要连接到备用 SpectroSERVER。

```
<ExportConfiguration
    export_devices           = "true"
    export_containers        = "true"
    export_port_attributes   = "false"
    export_links             = "true"
    export_topology_layout   = "true"
    export_annotation        = "true"
    export_WA_Link_models    = "true"
    export_spectrum_settings = "true"
    export_user_models       = "true"
    export_service_modeling  = "true"
    export_schedules         = "true"
    export_global_collections = "true"
    export_discovery_configs = "true"
    export_from_primary_ss_only = "true"
    export_policy_manager = "true"

/>
```

用于导出的 modelinggateway 工具

建模网关命令行工具“modelinggateway”（在 Windows 中为 modelinggateway.bat）位于 SS-Tools 目录中。其用于导出的语法是：

Windows

```
modelinggateway.bat -vnm vnm_name [-cmdb] -e export_file [-o outputfile] [-debug debugfile]
```

Solaris/Linux

```
modelinggateway -vnm vnm_name [-cmdb] -e export_file [-o outputfile] [-debug debugfile]
```

-vnm *vnm_name*

指定 SpectroSERVER 主机的名称。

-cmdb

（可选）将您 SpectroSERVER 的内容以 CA Spectrum 与 CA CMDB 进行集成时可以使用的格式导出。有关实施此集成的详细信息，请联系 CA 支持。

-e *export_file*

导出 CA Spectrum 拓扑数据。

-o *outputfile*

（可选）将错误信息记录到在 *outputfile* 参数中指定的文件。如果未使用此选项，则错误信息将记录到名为 *export_file*.log 的文件。*Export_file* 是 XML 文件的名称。

-debug *debugfile*

（可选）指示您要在导出过程中创建调试输出文件。使用 -debug 选项时，您可以为输出提供自己的调试文件名称。如果不提供 *debugfile* 的值，调试文件名称将默认为后缀为 .debug 的 *export_file* 名称。

注意：-debug 选项要求运行建模网关的计算机有磁盘空间。*export_file* 中模型的数量影响调试输出文件的大小：数据库中模型的数量越大，生成的调试文件就越大。

注意：要在另一台服务器上运行 modelinggateway 工具，请将 modelinggateway 工具及其所有支持文件移至该服务器。有关详细信息，请参阅 *《分布式 SpectroSERVER 管理员指南》*。

导出 CA Spectrum 拓扑数据

使用 modelinggateway 工具可导出 CA Spectrum 拓扑数据。

遵循这些步骤:

要导出 CA Spectrum 拓扑数据，请使用 -e 标志。例如，运行以下命令会将数据从 NOC1_Spectrum 上的 SpectroSERVER 导出到名为 NOC1_data.xml 的建模网关格式 XML 文件中：

```
modelinggateway -vnm NOC1_Spectrum -e NOC1_data.xml
```

导入建模网关 XML 文件

您可以将数据从建模网关格式的 XML 文件导入到 CA Spectrum。

遵循这些步骤:

要从建模网关格式的 XML 文件中导入数据，请使用 -i 标志。例如，运行以下命令会将数据从 NOC1_data.xml 导入到 NOC2_Spectrum 上的 SpectroSERVER。

```
modelinggateway -vnm NOC2_Spectrum -i NOC1_data.xml
```

附录 A： 文档类型定义元素

本部分介绍了文档类型定义 (DTD) 中定义的各元素的功能， 还提供了各元素的上下文。本部分*不*介绍 DTD 中使用的 XML 语法。有关语法的信息，请参阅 XML 参考。

Association

语法

父元素：

- Update
- Destroy

子元素：

- Left_Model
- Right_Model

规则：Association 元素必须包含一个 Left_Model 元素和一个 Right_Model 元素。

用法

Association 元素可创建或销毁模型之间的关联。

如果 Association 元素用作 Destroy 元素的子项，则将销毁指定的关联。
如果 Association 元素用作 Update 元素的子项，则将创建指定的关联。

属性

relation

指定此关联中 Left_Model 和 Right_Model 之间的 CA Spectrum 关系的名称或句柄。

Connection

语法

父元素：

- Topology
- Topology_Container
- Update
- Destroy

子元素：Device

规则：Connection 元素必须包含两个 Device 元素。

用法

Connection 元素指定两台设备之间的连接。Connection 元素必须始终包含两个 Device 元素，且其中每个 Device 元素都可包含零个或一个 Port 元素。如果指定了一个或两个端口，则连接在端口级别解析。如果没有指定一个或两个端口，则将触发发现以查找连接的一个或两个端口。

如果 Connection 元素用作 Destroy 元素的子项，则将销毁指定的连接。如果在其他任何上下文中使用 Connection，则将创建连接。

属性

create_pipe

指示指定连接的图形表示是否在 OneClick 中显示。

默认值：True

Correlation

语法

父元素：Import

子元素：Correlation_Domain

规则：Correlation 元素可以包含任意数量的子元素。

用法

Correlation 元素表示 Correlation Manager 模型，且与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

无。

Correlation_Domain

语法

父元素：Correlation

子元素：

- Device
- Port
- Model_Attr
- GenericView_Container

规则：Correlation_Domain 元素可以包含任意数量的子元素。

用法

Correlation_Domain 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A

CustomerManager

语法

父元素：SM_Service_Mgt

子元素：

- SM_Customer
- SM_CustomerGroup

规则：CustomerManager 元素可以包含任意数量的此类子元素。

用法

CustomerManager 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

注意：有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
containment_relation	字符	Groups_Customers	N/A
model_type	字符	CustomerManager	N/A

Destroy

语法

父元素：Import

子元素：

- Topology_Container
- Location_Container
- GenericView_Container
- Device

- Model
- Connection
- EventModel
- SM_Service
- SM_AttrMonitor
- SM_LatencyMon
- SM_ConnectMon
- SM_SLA
- SM_Guarantee
- SM_Customer
- 关联

规则：Destroy 元素可以根据需要包含任意数量的每个此类子元素。

使用情况

使用 Destroy 元素可删除容器模型、设备模型、连接和关联。您不能在 Destroy 元素中嵌套元素来表达层次结构或销毁层次结构。在 Destroy 元素中唯一允许的层次结构是 Connection-Device-Port 层次结构，它在端口级别指定要销毁的连接。您无需销毁包含在容器模型内的设备模型或其他容器模型，就可销毁该容器模型。在这种情况下，剩余的模型将放置在 CA Spectrum 的“Lost and Found”中。

属性

Destroy 元素没有任何属性。

Device

语法

父元素：

- Topology
- Location
- Topology_Container
- Left_Model
- Right_Model
- Location_Container

- GenericView
- GenericView_Container
- Connection
- Update
- Destroy
- SM_Service
- SM_AttrMonitor

子元素：

- Port
- Schedule

规则：

- **Port：**如果 Device 元素包含在 Connection 元素中，则仅允许一个 Port 元素。如果 Device 元素包含在用于更新端口的 Update 元素中，则允许多个 Port 元素。当 Device 元素包含在 View、Container 或 Destroy 元素中时，端口将被忽略。
- **Schedule：** Device 元素可以包含一个 Schedule 元素。

用法

要创建、销毁或更新设备模型，请使用 Device 元素。通过 Device 元素，您可以使用 IP 地址或 DNS 名称定义设备模型。

注意：如果在设备创建期间没有提供 community_string 和 agent_port 属性，则 CA Spectrum 会使用预定义的 SNMP 凭据创建设备。这些凭据在 OneClick 的 VNM 模型“信息”选项卡的“AutoDiscovery 控制”子视图中的“建模和协议选项”部分进行配置。

属性

ip_dnsname

指定设备的 IP 地址或 DNS 名称。如果设备不支持 SNMP 通信，您可以在此处将一个唯一字符串与指定的 model_type 一起使用。

secdomain_ipname

（可选）指定在设备所在安全域中运行 SDConnector 的主机的 IP 地址。

默认值：0.0.0.0

model_handle

（可选）指定用于标识现有设备模型的 model_handle。

注意：如果您提供了 model_handle，则会忽略 ip_dnsname 的值。

model_type

（可选）用于为您的设备建模的 CA Spectrum 模型类型。此设备模型可以是 .modelinggatewayresource.xml 文件中定义的任何智能模型类型。

注意：如果您已提供有效的 IP 地址或 DNS 名称，则不需要在此处指定值。

community_string

（可选）设备的团体字符串。

注意：如果不包括 community_string，则 CA Spectrum 使用第一个 SNMP 团体字符串值来创建设备。这些值在 OneClick 的 VNM 模型“信息”选项卡的“AutoDiscovery 控制”子视图中的“建模和协议选项”子视图中进行指定。

agent_port

（可选）控制与设备的 SNMP 代理进行通信时使用的端口号。

注意：如果不包括 agent_port，则 CA Spectrum 使用第一个 SNMP 端口值来创建设备。这些值在 OneClick 的 VNM 模型“信息”选项卡的“AutoDiscovery 控制”子视图中的“建模和协议选项”子视图中进行指定。

is_managed

（可选）当设置为 true 时，会将设备模型置于维护模式。

默认值：True

poll_interval

（可选）指定 SpectroSERVER 读取标记为 POLLED 的所有设备模型属性的时间间隔（秒）。

log_ration

（可选）指定在数据库中记录轮询结果之前 SpectroSERVER 对设备进行轮询的次数。

poll_status

（可选）允许您通过将轮询状态设置为 false 来禁止 SpectroSERVER 对设备的轮询。

model_name

（可选）指定模型的名称。

DeviceType

（可选）指定模型的设备类型。

注意：有关设备类型的详细信息，请参阅《*认证用户指南*》。

reconfig

（可选）指定建模网关是否向 SpectroSERVER 发送操作以重新配置设备模型。

discover_connections

（可选）当设置为 true 时，会对任何新创建的设备模型运行发现以自动映射模型连接。

EventModel

语法

父元素：

- Topology_Container
- Left_Model
- Right_Model

子元素：无

规则：N/A

用法

要导入 EventModel 模型以便与 Southbound Gateway 集成一起使用，请使用 EventModel 元素。有关 Southbound Gateway 的详细信息，请参阅《*Southbound Gateway Toolkit Guide*》（Southbound Gateway 工具包指南）。

属性

model_name

指定已实例化或标识的模型的唯一名称。

unique_id

指定用于唯一定义此 EventModel 模型所表示的事件源的标识符。有关详细信息，请参阅《*Southbound Gateway Toolkit Guide*》（Southbound Gateway 工具包指南）。

model_handle

（可选）指定用于标识现有设备模型的 model_handle。

Security_String

（可选）指定 EventModel 的安全字符串。

默认值：public

manager_name

（可选）指定正在使用 Southbound Gateway 的第三方应用程序的名称。

注意：对于任何此处未列出的应用程序，请使用默认值。

默认值：0

1

NetMentor

2

SSM

3

Omni2000

GenericView

语法

父元素：Import

子元素：

- GenericView_Container
- Device

规则：GenericView 元素可以包含任意数量的此类子元素。

用法

要创建除拓扑视图和位置视图之外的自定义分层视图，请使用 GenericView 元素。您可以修改此元素以满足您的集成需求。

属性

containment_relation

指定用于定义由哪个 CA Spectrum 关系来定义此视图中的遏制关系的关系句柄。

限制：必须是 CA Spectrum 遏制关系。

model_type

指定用于表示为此视图定义的顶层容器模型的模型类型。此 model_type 必须使用其在 .modelinggatewayresource.xml 文件中的模型句柄来指定。

name

指定在 GenericView 层次结构顶层的实例化容器模型的唯一名称。

complete_topology

（可选）当设置为 true 时，销毁 GenericView 视图中任何未指定的现有容器和设备模型。此外，还销毁该视图的子容器中的此类模型。

GenericView_Container

语法

父元素：GenericView

子元素：

- GenericView_Container
- Device

规则：GenericView_Container 元素可以包含任意数量的子元素。

用法

要在常规视图中创建容器模型，请使用 GenericView_Container 元素。GenericView 元素和 GenericView_Container 元素都可用于创建自定义视图。因此，您作为集成人员，需要决定什么情况下使用此容器以及如何使用。

属性

name

指定已实例化或标识的模型的名称。需要使用 model_type 和 name 属性来唯一标识 GenericView_Container。默认情况下，此属性用于设置 CA Spectrum 模型名称属性 (attr id 0x1006e) 的值。但是，此属性可以更改为 .modelinggatewayresource.xml 文件中的任何其他属性。您可以更改 .modelinggatewayresource.xml，以便名称映射到不同的属性。在这种情况下，该新属性（与模型类型一起）用于标识容器。此行为使两个容器具有相同的模型名称。

model_type

指定用于创建模型的 CA Spectrum 模型类型。此 model_type 必须使用其在 .modelinggatewayresource.xml 文件中的模型句柄来指定。需要使用 model_type 和 name 属性来唯一标识 GenericView_Container。

containment_relation

（可选）在 Generic_Container 和容器内的模型之间存在的 CA Spectrum 关系的名称。如果没有为此属性指定值，则继承父模型的遏制关系。

GlobalCollection

语法

父元素： Import

子元素：

- Device
- Topology_Container
- Location_Container

用法

表示 GlobalCollection 模型。

属性

name

指定此全局集合的名称。

containment_relation

指定用于定义由哪个 CA Spectrum 关系来定义此视图中的遏制关系的关系句柄。

默认值：“GlobalCollect”

collectionDescription

（可选）描述全局集合。

Security_String

（可选）指定全局集合的安全字符串。

Import

语法

父元素：无

子元素：

- Topology
- Location
- GenericView
- Update
- Destroy
- SM_Service_Mgt
- Correlation
- GlobalCollection

规则：Import 元素可以包含每个此类子元素中的一个子元素。

用法

Import 元素是根元素，且必须包含在每个输入文件中。

属性

model_activation_time

指定允许用于每次激活设备模型的最大分钟数。

数据类型：字符

默认值：5 分钟

Left_Model

语法

父元素：Association

子元素：

- Device
- Port
- Topology_Container
- Location_Container
- EventModel
- Model

规则：Left_Model 元素可以仅包含一个子元素。

用法

Left_Model 元素定义关联中的左侧模型。

属性

无。

List_Value

语法

父元素：Model_Attr

子元素：无

规则：N/A

用法

要指定 CA Spectrum 列表属性值，请使用 List_Value 元素。

属性

无。

位置

语法

父元素： Import

子元素：

- Location_Container
- Device

规则： Location 元素可以包含任意数量的子元素。

用法

要指定您要在 OneClick 的位置视图（World 拓扑）中创建模型，请使用 Location 元素。

属性

complete_topology

（可选）当设置为 true 时，在导入期间会销毁位置视图或任何子容器中任何未指定的现有容器和设备模型。

默认值： False

数据类型： 布尔值

Location_Container

语法

父元素：

- Location
- Location_Container
- Left_Model

- Right_Model
- GlobalCollection

子元素：

- Location_Container
- Device

规则：Location_Container 元素可以包含任意数量的子元素。

用法

要创建或指定用于对位置视图中的模型和其他位置容器进行分组的 Location_Container 模型，请使用 Location_Container 元素。

属性

name

已实例化或标识的模型的名称。需要使用 model_type 和 name 属性来唯一标识 Location_Container。
默认情况下，此属性用于设置 CA Spectrum 模型名称属性 (attr id 0x1006e) 的值。但是，此属性可以更改
为 .modelinggatewayresource.xml 文件中的任何其他属性。您可以更改 .modelinggatewayresource.xml，以便名称映射到不同的属性。在这种情况下，该新属性（与模型类型一起）用于标识容器。此行为使两个容器具有相同的模型名称。

model_type

指示您要创建的模型的类型。需要使用 model_type 和 name 属性来唯一标识 Location_Container。可能的值包括：

- Country
- Region
- Site
- Building
- Floor
- Section
- Room

model_handle

（可选）可用于标识模型。如果您提供了 model_handle，则会忽略 name 和 model_type 的值。

Security_String

（可选）定义 CA Spectrum 用户访问模型的要求。每个安全字符串包括一个或多个安全团体条目并分配至模型。

model_name

（可选）要更改模型的名称，请使用 name 和 model_name 属性。name 属性指定旧名称，而 model_name 属性指定新名称。

model_modify_author

（可选）将数据写入 CA Spectrum 属性 mdl_modfy_athr。

complete_topology

（可选）当设置为 true 时，在导入期间会销毁位置视图或任何子容器中任何未指定的现有容器和设备模型。

默认值： False

Model_Attr

语法

父元素： Correlation_Domain

子元素： List_Value

规则： Model_Attr 元素可以包含任意数量的子元素。

用法

要指定其值包含多行文本或值列表的 CA Spectrum 属性，请使用 Model_Attr 元素。

属性

attr_id

指示您要指定的属性的 CA Spectrum 属性 ID。

数据类型： 字符

Model

语法

父元素：

- Left_Model
- Right_Model

子元素：无

用法

要表示任何 CA Spectrum 模型，请使用 Model 元素。

属性

name

指定此模型的名称。

model_type

指定此模型的模型类型。

model_handle

（可选）指定此模型的 model_handle。如果指定了 model_handle 值，则会忽略 name 和 model_type 的值。

MonitorPolicy_Attr

语法

父元素：

- SM_Service
- SM_AttrMonitor

子元素：无

规则：N/A

用法

MonitorPolicy_Attr 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

无。

Port

语法

父元素：

- Device
- Left_Model
- Right_Model
- Correlation_Domain
- SM_Service
- SM_AttrMonitor

子元素：无

规则：N/A

用法

Port 元素用于指定端口级别的连接或更新端口属性。更新时，父元素 Device 包含在 Update 元素中。指定连接时，父元素 Device 包含在 Connection 元素中。

属性

identifier_name

与 identifier_value 属性一起使用来唯一标识端口。identifier_name 可以是在“可能值”列中列出的任何一个 MIB OID 名称。portID 值可用于通过端口的 Component_OID 属性 (0x1006a) 来标识端口。如果 Port 元素表示帧中继虚拟电路，请使用 frCircuitTableInstance。如果 Port 元素表示 ATM 虚拟信道或路径链路，请使用 atmVclTableInstance。

portID 值可用于通过端口的 Component_OID 属性 (0x1006a) 来标识端口。如果 Port 元素表示帧中继虚拟电路，请使用 frCircuitTableInstance。如果 Port 元素表示 ATM 虚拟信道或路径链路，请使用 atmVclTableInstance。可能的值包括：

- ifIndex
- ipAddress
- ifPhysAddress

- ifName
- ifAlias
- model_name
- portDescription
- portID
- frCircuitTableInstance
- atmVclTableInstance
- atmVplTableInstance

identifier_value

指定 identifier_name 选项的值。

model_handle

（可选）指定用于标识现有模型的 model_handle。

注意：如果您提供了 model_handle，则会忽略 identifier_name 和 identifier_value 的值。

ip_dnsname

（可选）指定端口模型的 IP 地址或 DNS 名称。如果端口模型不支持 SNMP 通信，您可以在此处将一个唯一字符串与指定的 model_type 一起使用。

model_name

（可选）指定您要更新的模型的名称。

circuit_id

（可选）使用 ID 标识在 ATM 或帧中继连接中涉及的电路。

circuit_name

（可选）使用名称标识在 ATM 或帧中继连接中涉及的电路。

log_ratio

（可选）指定在数据库中记录轮询结果之前发生的端口模型轮询次数。

poll_interval

（可选）指定 SpectroSERVER 读取标记为 POLLED 的所有端口模型属性的时间间隔（秒）。

poll_status

（可选）允许管理员通过将轮询状态设置为 False 来禁用端口模型轮询。

Right_Model

语法

父元素： Association

子元素：

- Device
- Port
- Topology_Container
- Location_Container
- EventModel
- Model

规则： Right_Model 元素可以仅包含一个子元素。

用法

Right_Model 元素定义关联中的右侧模型。

属性

无。

RTM_Test

语法

父元素：

- SM_Service
- SM_AttrMonitor

子元素： 无

规则： N/A

用法

RTM_Test 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
model_type	字符	RTM_Test	N/A

Schedule

语法

父元素： Device

子元素： 无

用法

要为特定设备模型创建维护模式排定，请使用 Schedule 元素。

属性

name

指定排定的名称。

SCHED_Recurrence

指定设备模型置于维护模式的频率。

1

始终 (24x7)

2

按天

3

按周

4

按月

5

按年

默认值： 1

SCHED_Start_Hour

（可选）指定设备置于维护模式的小时。

限制：0-23

SCHED_Start_Minute

（可选）指定设备置于维护模式的分钟。

限制：0-59

SCHED_Start_DoW

（可选）指定设备要在星期几置于维护模式。

0

星期日

1

星期一

2

星期二

3

星期三

4

星期四

5

星期五

6

星期六

SCHED_Start_DoM

（可选）指定设备置于维护模式的当月日期。

限制：1-31

SCHED_Start_Month

（可选）指定设备置于维护模式的月份。

0

一月

1

二月

2

三月

3

四月

4

五月

5

六月

6

七月

7

八月

8

九月

9

十月

10

十一月

11

十二月

SCHED_Duration

（可选）指定设备置于维护模式的时间长度（秒）。

默认值：0

SCHED_Recurrence_Multiplier

（可选）指定确定每次排定维护模式开始之间的时间长度的重复单位（天、周、月、年）数。

默认值：1

SCHED_Daily_Repeat_Limit

（可选）指定在每个重复周期中重复排定维护（由 SCHED_Start_Hour 和 SCHED_Start_Minute 指定）的连续天数。此属性仅适用于按周、按月或按年的重复。

SM_AttrMonitor

语法

父元素：SM_Service

子元素：无

规则：N/A

用法

SM_AttrMonitor 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
containment_relation	字符	N/A	SLMMonitors SLMWatchesContainer
AttrToWatch	字符	N/A	N/A
MonitorPolicy_ID	字符	N/A	N/A
is_managed	布尔值	N/A	True False
Generate_Service_Alarms	布尔值	N/A	True False
model_type	字符	SM_AttrMonitor	N/A

SM_Customer

语法

父元素：

- SM_CustomerGroup
- CustomerManager

子元素：无

规则：N/A

用法

SM_Customer 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
containment_relation	字符	N/A	SlmAgreesTo SlmUses
Security_String	字符	N/A	N/A
CustomerID	字符	N/A	N/A
Criticality	字符	N/A	N/A
CustomerField4	字符	N/A	N/A
CustomerField5	字符	N/A	N/A
CustomerField6	字符	N/A	N/A
CustomerField7	字符	N/A	N/A
Contact_Name	字符	N/A	N/A

属性	数据类型	默认值	可能值
Contact_Title	字符	N/A	N/A
Contact_Location	字符	N/A	N/A
Email_Address	字符	N/A	N/A
Phone_Number	字符	N/A	N/A
Mobile_Phone_Number	字符	N/A	N/A
Pager_Number	字符	N/A	N/A
Fax_Number	字符	N/A	N/A
User_Defined_1	字符	N/A	N/A
User_Defined_2	字符	N/A	N/A
User_Defined_3	字符	N/A	N/A
User_Defined_4	字符	N/A	N/A
Secondary_Contact_Name	字符	N/A	N/A
Secondary_Contact_Location	字符	N/A	N/A
Secondary_Email_Address	字符	N/A	N/A
Secondary_Phone_Number	字符	N/A	N/A
Secondary_Mobile_Phone_ 编号	字符	N/A	N/A
Secondary_Pager_Number	字符	N/A	N/A
Secondary_Fax_Number	字符	N/A	N/A
Secondary_User_Defined_1	字符	N/A	N/A
Secondary_User_Defined_2	字符	N/A	N/A

属性	数据类型	默认值	可能值
Secondary_User_Defined_3	字符	N/A	N/A
Secondary_User_Defined_4	字符	N/A	N/A
model_type	字符	SM_Customer	N/A

SM_CustomerGroup

语法

父元素：

- CustomerManager
- SM_CustomerGroup

子元素：

- SM_CustomerGroup
- SM_Customer

规则：SM_CustomerGroup 元素可以包含任意数量的此类子元素。

用法

SM_CustomerGroup 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	NA	N/A
containment_relation	字符	Groups_Customer	N/A
model_type	字符	SM_CustomerGroup	N/A

SM_Guarantee

语法

父元素：SM_SLA

子元素：无

规则：N/A

用法

SM_Guarantee 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
containment_relation	字符	SlmIsMeasuredBy	N/A
is_managed	布尔值	N/A	True False
DegradedTimeViolationLevel	字符	N/A	N/A
DegradedTimeWarningLevel	字符	N/A	N/A
DownTimeViolationLevel	字符	N/A	N/A
DownTimeWarningLevel	字符	N/A	N/A
LorTimeViolationLevel	字符	N/A	N/A
LorTimeWarningLevel	字符	N/A	N/A
model_type	字符	SM_Guarantee	N/A

SM_LatencyMon

语法

父元素：

- SM_Guarantee
- SM_AttrMonitor
- SM_Service

子元素：

- Topology_Container
- MonitorPolicy_Attr

规则：SM_LatencyMon 元素可以包含任意数量的此类子元素。

用法

SM_LatencyMon 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
containment_relation	字符	N/A	SlmMonitors SlmWatchesContainer
is_managed	布尔值	N/A	True False
DefaultMaxRTT	字符	N/A	N/A
DefaultMeasureInterval	字符	N/A	N/A
mode_type	字符	SM_LatencyMon	N/A

SM_Service

语法

父元素：

- SM_Service
- SM_ServiceMgr

子元素：

- SM_Service
- SM_AttrMonitor

规则：SM_Service 元素可以包含任意数量的此类子元素。

用法

SM_Service 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
Criticality	字符	N/A	N/A
containment_relation	字符	N/A	SlmMonitors SlmWatchesContainer
AttrToWatch	字符	N/A	N/A
MonitorPolicy_ID	字符	N/A	N/A
is_managed	布尔值	N/A	True False
Generate_Service_Alarms	布尔值	N/A	True False
Security_String	字符	N/A	N/A

属性	数据类型	默认值	可能值
model_type	字符	SM_Service	N/A

SM_Service_Mgt

语法

父元素：Import

子元素：

- SM_ServiceMgr
- CustomerManager
- SM_SLA_Mgr

规则：仅每个此类子元素的单个实例可以存在于 SM_Service_Mgt 中。

用法

SM_Service_Mgt 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	服务管理	N/A
containment_relation	字符		N/A
model_type	字符		N/A

SM_ServiceMgr

语法

父元素：SM_Service_Mgt

子元素：SM_Service

规则：SM_ServiceMgr 元素可以包含任意数量的此子元素。

用法

SM_ServiceMgr 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name	字符	N/A	N/A
containment_relation	字符	SlmContains	N/A
model_type	字符	SM_ServiceMgr	N/A

SM_SLA

语法

父元素：SM_SLA_Mgr

子元素：SM_Guarantee

规则：SM_SLA 元素可以包含任意数量的此子元素。

用法

SM_SLA 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name *（必需）*	字符	N/A	N/A
containment_relation	字符	SlmContains	N/A
is_managed	布尔值	N/A	True False
Security_String	字符	N/A	N/A
model_type	字符	SM_SLA	N/A

SM_SLA_Mgr

语法

父元素：SM_Service_Mgt

子元素：SM_SLA

规则：SM_SLA_Mgr 元素可以包含任意数量的此子元素。

用法

SM_SLA_Mgr 元素与 CA Spectrum Service Manager 一起使用。有关用法详细信息，请参阅《*Service Manager 用户指南*》。

属性

有关属性定义，请参阅《*Service Manager 用户指南*》。

属性	数据类型	默认值	可能值
name	字符		N/A
containment_relation	字符	SlmContains	N/A
model_type	字符	SM_ServiceMgr	N/A

拓扑

语法

父元素： Import

子元素：

- Topology_Container
- Device
- Connection

规则： Topology 元素可以包含任意数量的此类子元素。

用法

要在 OneClick 拓扑视图（Universe 拓扑）中创建模型，请使用 Topology 元素。

属性

complete_topology

当设置为 true 时，在导入期间销毁拓扑视图中任何未指定的现有容器和设备模型。此外，还销毁该视图的任何子容器。

默认值： False

discover_connections

当设置为 true 时，对任何新创建的设备模型运行发现以自动映射模型连接。

默认值： False

Topology_Container

语法

父元素：

- Topology
- Topology_Container
- SM_Service
- SM_AttrMonitor

子元素：

- Topology_Container
- Device
- EventModel
- Connection

规则：Topology_Container 元素可以包含任意数量的此类子元素。

用法

要创建或指定用于对拓扑视图中的模型和其他拓扑容器进行分组的 Topology_Container 模型，请使用 Topology_Container 元素。可能的模型类型在下表的 model_type 部分中列出。

属性

model_type

指示您要创建的模型的类型。需要使用 model_type 和 name 属性以唯一标识 Topology_Container。

- Network
- LAN
- IPClassA
- IPClassB
- IPClassC
- LAN_802_3
- LAN_803_5
- EventAdmin
- ATM_Network

model_handle

（可选）可用于标识现有模型。如果您提供了 model_handle，则会忽略 model_type 和 model_name 的值。

Security_String

（可选）指定分配给模型的 CA Spectrum 安全级别。

subnet_address

（可选）指定设备的子网地址。

subnet_mask

（可选）指定确定设备 IP 地址所属子网的掩码。

model_name

（可选）指定容器模型的模型名称。

trapIPAddress

（可选）仅用于 EventAdmin 模型。

x_coordinate

（可选）指定模型在拓扑中的 x 坐标。

y_coordinate

（可选）指定模型在拓扑中的 y 坐标。

complete_topology

（可选）当设置为 True 时，在导入期间会销毁此 Topology_Container 和任何子容器中任何未指定的现有容器和设备模型。

默认值： False

discover_connections

（可选）指定 CA Spectrum 是否发现连接到此模型的设备并为其建模。

Update

语法

父元素： Import

子元素：

- Topology_Container
- Location_Container
- GenericView_Container
- Connection
- Device
- Model
- EventModel
- SM_Service
- SM_AttrMonitor
- SM_LatencyMon
- SM_ConnectMon

- SM_SLA
- SM_Guarantee
- SM_Customer
- Association

规则：Update 元素可以包含任意数量的此类子元素。

用法

要更新任何设备、容器或端口子元素的属性，请使用 Update 元素。

注意：使用此元素时不允许分层指定，但在 Device 元素中使用 Port 元素除外。

属性

无。

附录 B： 文档类型定义文件

本部分包含文档类型定义 (DTD)，它定义用于导入的 XML 元素和属性。

注意：以下代码可能不是该文件的最新版本。有关最新版本的 DTD，请使用随建模网关工具包提供的实际文件。该文件位于 SS-Tools 目录中，且名为 .modelinggateway.dtd。

```
<!-- ********************************************************** -->
<!-- 根 Import 元素包含 Topology 的 0 或 1          -->
<!-- Location、GenericView、Update、Destroy 元素、  -->
<!-- SM_Service_Mgt、Correlation 和全局集合。-->
<!--                                                            -->
<!-- 该元素具有一个属性 model_activation_time，      -->
<!-- 它是每个设备模型激活时等待的最长时间，          -->
<!-- 以分钟为单位来表示。默认值为 5 分钟。-->
<!-- ********************************************************** -->

<!ELEMENT Import ( ( Topology       |
                     Location       |
                     GenericView    |
                     Update         |
                     Destroy        |
                     SM_Service_Mgt |
                     Correlation    |
                     GlobalCollection )*) >

<!ATTLIST Import model_activation_time  CDATA     "5">

<!-- ************************************************** -->
<!-- 用于“拓扑”视图的 Topology 元素，  -->
<!-- 包含任意数量的 Topology_Container、      -->
<!-- Device 和 Connection 元素。-->
<!--                                                    -->
<!-- 此元素有两个属性：complete_topology   -->
<!-- 和 discover_connection。-->
<!--                                                    -->
<!-- ************************************************** -->

<!ELEMENT Topology ((Topology_Container | Device | Connection)*) >
<!ATTLIST Topology
        complete_topology    (false | true)     #IMPLIED
        discover_connections (false | true)     #IMPLIED>

<!-- ************************************************** -->
<!-- 用于拓扑容器模型的 Topology_Container 元素  -->
<!-- 包含任意数量的      -->
<!-- Topology_Container、Device 和 Connection 元素。-->
<!--                                                    -->
```

```
<!-- “model_type”和“name”是唯一标识  -->
<!-- Topology_Container 模型的必需属性。-->
<!--                                                    -->
<!-- “model_handle”也可用于标识模型。-->
<!-- 如果提供了“model_handle”，则会忽略“name”  -->
<!-- 和“model_type”的值。-->
<!--                                                    -->
<!-- “trapIPAddress”属性应仅用于     -->
<!-- EventAdmin 模型。-->
<!--                                                    -->
<!-- ************************************************** -->

<!ELEMENT Topology_Container ((Topology_Container |
                               Device |
                               EventModel |
                               Connection )*) >

<!ATTLIST Topology_Container
        name          CDATA                  #REQUIRED
        model_type          ( Network      |
                              Lan          |
                              IPClassA     |
                              IPClassB     |
                              IPClassC     |
                              LAN_802_3    |
                              LAN_803_5    |
                              EventAdmin   |
                              ATM_Network  ) #REQUIRED

        model_handle          CDATA          #IMPLIED
        Security_String       CDATA          #IMPLIED
        subnet_address        CDATA          #IMPLIED
        subnet_mask           CDATA          #IMPLIED
        model_name            CDATA          #IMPLIED
        trapIPAddress         CDATA          #IMPLIED
        x_coordinate          CDATA          #IMPLIED
        y_coordinate          CDATA          #IMPLIED
        complete_topology    (false | true)  #IMPLIED
        discover_connections (false | true)  #IMPLIED >

<!-- ************************************************** -->
<!-- 用于“位置”视图的 Location 元素          -->
<!-- 包含任意数量的 Location_Container 和          -->
<!-- Device 元素                                        -->
<!--                                                    -->
<!-- 该元素具有 complete_topology 属性。-->
<!-- ************************************************** -->

<!ELEMENT Location ( (Location_Container  | Device )* ) >
<!ATTLIST Location
        complete_topology    (false | true)     #IMPLIED>
```

```
<!-- ************************************************** -->
<!-- 用于位置容器的 Location_Container 元素  -->
<!-- Location 容器可以包含任意数量的          -->
<!-- Location_Container 和 Device 元素。-->
<!--                                                    -->
<!-- “model_type”和“name”是唯一标识  -->
<!-- Location_Container 模型的必需属性。-->
<!--                                                    -->
<!-- “model_handle”也可用于标识模型。-->
<!-- 如果提供了“model_handle”，则会忽略“name”  -->
<!-- 和“model_type”的值。-->
<!-- ************************************************** -->

<!ELEMENT Location_Container ( ( Location_Container | Device )* )>

<!ATTLIST Location_Container
        name          CDATA                  #REQUIRED
        model_type        ( Country     |
                        Region      |
                        Site        |
                        Building    |
                        Floor       |
                        Section     |
                        Room        )     #REQUIRED

        model_handle          CDATA          #IMPLIED
        Security_String       CDATA          #IMPLIED
        model_name            CDATA          #IMPLIED
        model_modify_author CDATA            #IMPLIED
        complete_topology   (false | true)   #IMPLIED >

<!-- ********************************************************** -->
<!-- Device 元素用于设备模型。-->
<!--                                                            -->
<!-- “ip_dnsname”是唯一标识设备模型的        -->
<!-- 必需属性。-->
<!--                                                            -->
<!-- “model_handle”也可用于标识设备模型。-->
<!-- 如果提供了“model_handle”，则会忽略“ip_dnsname”  -->
<!-- 的值。-->
<!--                                                            -->
<!-- 注意:                                                   -->
<!--                                                            -->
<!-- 1. 如果属性 is_managed 设为 false，则设备   -->
<!--    不可联系。因此，您必须设置 model_type-->
<!--    属性。-->
<!--                                                            -->
<!-- 2. 一个 Device 可以分别根据以下情况包含 0、1      -->
<!--    或多个 Port:                        -->
<!--                                                            -->
<!-- (a) 如果 Device 在 Container 或 Destroy 元素中,       -->
<!--     则不需要 Port 元素。如果提供了 Port       -->
<!--     元素，它们将被忽略。-->
<!--                                                            -->
```

```
<!-- (b) 如果 Device 是在 Connection 元素中，则只允许一个 Port    -->
<!--     元素。-->
<!--                                                              -->
<!-- (c) 如果 Device 包含在用于更新端口的 Update 元素中，   -->
<!--     则允许多个 Port 元素。-->
<!--                                                              -->
<!-- (d) 如果在设备上指定 discover_connections="true"   -->
<!--     则不必再在该设备的父容器上指定       -->
<!--     容器。在容器上指定该属性    -->
<!--     会导致 Spectrum 在该容器中的        -->
<!--     每个模型上发现连接，      -->
<!--     因此会出现性能和效率方面的问题。-->
<!--                                                              -->
<!-- ************************************************************ -->

<!ELEMENT Device ( Port* | Schedule ) >

<!ATTLIST Device
        ip_dnsname        CDATA             #REQUIRED
        secdomain_ipname  CDATA             #IMPLIED

        model_handle          CDATA         #IMPLIED
        model_type        CDATA             #IMPLIED
        community_string  CDATA             #IMPLIED
       agent_port        CDATA             #IMPLIED
        poll_interval     CDATA             #IMPLIED
        log_ratio         CDATA             #IMPLIED
        model_name            CDATA         #IMPLIED
        DeviceType            CDATA         #IMPLIED
       x_coordinate          CDATA          #IMPLIED
       y_coordinate          CDATA          #IMPLIED
        is_managed        (true  | false)    #IMPLIED
        reconfig          (true  | false)    #IMPLIED
        poll_status       (true  | false)    #IMPLIED
        discover_connections (false | true)  #IMPLIED >

<!-- ************************************************************ -->
<!-- Port 元素用于设备端口模型。-->
<!--                                                              -->
<!-- “identifier_name”和“identifier_value”是     -->
<!-- 唯一标识端口模型的必需属性。-->
<!--                                                              -->
<!-- “model_handle”也可用于标识端口模型。-->
<!-- 如果提供了“model_handle”，则会忽略 identifier_name  -->
<!-- 和 identifier_value 的值。-->
<!-- ************************************************************ -->

<!ELEMENT Port ( Port* ) >

<!ATTLIST Port
        identifier_name ( portDescription        |
                          model_name             |
                          ifIndex                |
```

```
                        ipAddress              |
                        ifPhysAddress          |
                        ifName                 |
                        ifAlias                |
                        portID                 |
                        frCircuitTableInstance |
                        atmVclTableInstance    |
                        atmVplTableInstance    ) #REQUIRED
        identifier_value  CDATA                  #REQUIRED

        model_handle      CDATA                  #IMPLIED
        ip_dnsname        CDATA                  #IMPLIED
        model_type        CDATA                  #IMPLIED
        model_name        CDATA                  #IMPLIED
        circuit_id        CDATA                  #IMPLIED
        circuit_name      CDATA                  #IMPLIED
        log_ratio         CDATA                  #IMPLIED
        poll_interval     CDATA                  #IMPLIED
        poll_status       (false | true)         #IMPLIED >

<!-- ************************************************************************
-->
<!-- 表示 Schedule 模型                                                    -->
<!--                                                                        -->
<!-- SCHED_Recurrence 可具有以下值                                          -->
<!--                                                                        -->
<!--  1 = 始终 (24 x 7)                                                     -->
<!--  2 = 每天                                                              -->
<!--  3 = 每周                                                              -->
<!--  4 = 每月                                                              -->
<!--  5 = 每年                                                              -->
<!--  6 = 一次                                                              -->
<!--                                                                        -->
<!--  SCHED_Start_Hour：值的范围为 0-23                                     -->
<!--  SCHED_Start_Minute：值的范围为 0-59                                   -->
<!--  SCHED_Start_DoW：每周重复的周日期     -->
<!--                （值的范围为 0-6，其中周日为 0）                        -->
<!--  SCHED_Start_DoM：每月和每年重复的月份日期 -->
<!--                   （值的范围为 1-31）                                  -->
<!--  SCHED_Start_Month：每年重复的月份     -->
<!--                   （值的范围为 0-11）                                  -->
<!--  SCHED_Duration：活动时段时长（秒）。可能为 0（默认值） -->
<!--  SCHED_Recurrence_Multiplier：指定确定每次排定维护模式          -->
<!--                               开始之间的时间长度 -->
<!--                               的重复单位（天、周、月、年）数。-->
<!--                               默认值为 1。-->
```

```
<!--  SCHED_Daily_Repeat_Limit：从每个重复周期开始       -->
<!--                            重复每日排定（用                -->
<!--                             SCHED_Start_Hour 和 SCHED_Start_Minute 指定）
的    -->
<!--                            连续天数。-->
<!--                           仅适用于每周、每月或        -->
<!--                            每年重复。-->
<!-- SCHED_DayBitMask：“每周排定”应为“活动”的       -->
<!--                   周日期。值为：                          -->
<!--                   星期日 = 1，                                          -->
<!--                   星期一 = 2，                                          -->
<!--                   星期二 = 4，                                          -->
<!--                   星期三 = 8，                                          -->
<!--                   星期四 = 16，                                        -->
<!--                   星期五 = 32，                                         -->
<!--                   星期六 = 64，                                         -->
<!--                    例如，如果需要星期一、星期三和星期五，则值应为    -->
<!--                   2+8+32=42                                             -->
<!--  SCHED_Start_MoY、                                                      -->
<!--  SCHED_START_YEAR、                                                     -->
<!--  SCHED_START_DAY：与 SCHED_START_MONTH 配合使用         -->
<!--                   表示排定应该在未来的某些天内处于     -->
<!--                   活动状态。SCHED_START_YEAR 和         -->
<!--                   SCHED_START_DAY 必须为非零值。-->
<!--                   否则，排定会正常运行，并在当天尽可能早的       -->
<!--                   时候变为激活。-->
<!--                   请注意，应该将 SCHED_START_YEAR 指定为     -->
<!--                   自 1900 年起的年数。-->
<!-- SCHED_Description：该排定的说明。-->
<!--                                                                        -->
<!-- ************************************************************************
-->

<!ELEMENT Schedule ( #PCDATA) >

<!ATTLIST Schedule
        name                    CDATA                   #REQUIRED
        SCHED_Recurrence            ( 1 | 2 | 3 |
                                        4 | 5 | 6 )        #REQUIRED
        SCHED_Daily_Repeat_Limit    CDATA               #REQUIRED
        SCHED_Duration              CDATA               #REQUIRED
        SCHED_Recurrence_Multiplier CDATA               #REQUIRED
        SCHED_Start_DoM             CDATA               #REQUIRED
        SCHED_Start_DoW             CDATA               #REQUIRED
        SCHED_Start_Hour            CDATA               #REQUIRED
        SCHED_Start_Minute          CDATA               #REQUIRED
        SCHED_Start_Month           CDATA               #REQUIRED
        SCHED_Start_Day             CDATA               #REQUIRED
        SCHED_DayBitMask            CDATA               #REQUIRED
        SCHED_Start_Year            CDATA               #REQUIRED
```

```
        SCHED_Start_MoY             CDATA               #REQUIRED
        SCHED_Description           CDATA               #REQUIRED
        >
<!-- ************************************************************ -->
<!-- 用于表示 EventModels 的元素。-->
<!--                                                              -->
<!-- 出于性能原因，必须指定唯一 id。-->
<!-- ************************************************************ -->

<!ELEMENT EventModel ( #PCDATA ) >

<!ATTLIST EventModel
        model_name          CDATA       #REQUIRED
        unique_id           CDATA       #REQUIRED
        model_handle        CDATA       #IMPLIED
        Security_String     CDATA       "public"
        manager_name        CDATA       "0">

<!-- ************************************************************ -->
<!-- Connection 元素表示设备连接。-->
<!-- 它包含连接中涉及的两个 Device 元素。-->
<!-- 每个 Device 可能包含 0 或 1 个 Port 元素。-->
<!--                                                              -->
<!-- Connection 有一个 create_pipe 属性。通常，对于 ATM    -->
<!-- 电路链路，将 create_pipe 设为 false 以避免        -->
<!-- 在视图中创建太多管道。在这种情况下，        -->
<!-- 可以使用 ATM 管理器来查看连接。由        -->
<!-- 用户决定 create_pipe 的设置。默认情况下，     -->
<!-- 将为每个连接创建一个管道。-->
<!--                                                              -->
<!-- ************************************************************ -->

<!ELEMENT Connection (Device, Device)>

<!ATTLIST Connection  create_pipe  (true | false)  "true" >

<!-- ************************************************************ -->
<!-- Update 元素用于更新 SPECTRUM 模型属性和     -->
<!-- 关联。-->
<!--                                                              -->
<!-- Update 元素可以包含任意数量的        -->
<!-- 要更新的 Container、Device 和 Association 元素。-->
<!-- 要更新端口，Port 元素需要放置在   -->
<!-- Device 元素中，然后将 Device 元素放置在     -->
<!-- Update 元素中。-->
<!-- ************************************************************ -->
```

```
<!ELEMENT Update  ( ( Topology_Container          |
                     Location_Container          |
                     GenericView_Container       |
                     Connection                  |
                     Device                      |
                     EventModel                  |
                     SM_Service                  |
                     SM_AttrMonitor              |
                     SM_LatencyMon               |
                     SM_ConnectMon               |
                     SM_SLA                      |
                     SM_Guarantee                |
                     SM_Customer                 |
                     Association
                   )* ) >

<!-- ********************************************************** -->
<!-- Destroy 元素：销毁 SPECTRUM 模型和关联。-->
<!--                                                            -->
<!-- Destroy 元素可以包含任意数量的       -->
<!-- 要销毁的 Container、Device、Connection    -->
<!-- 和 Association 元素。-->
<!-- ********************************************************** -->

<!ELEMENT Destroy ( ( Topology_Container          |
                     Location_Container          |
                     GenericView_Container       |
                     Device                      |
                     Connection                  |
                     EventModel                  |
                     SM_Service                  |
                     SM_AttrMonitor              |
                     SM_LatencyMon               |
                     SM_ConnectMon               |
                     SM_SLA                      |
                     SM_Guarantee                |
                     SM_Customer                 |
                     Association

                   )* ) >
```

```
<!-- ********************************************************** -->
<!-- Association 元素定义要创建或销毁的 -->
<!-- 两个模型之间的 SPECTRUM 关联。-->
<!--                                                            -->
<!-- Association 元素包含一个 Left_Model 元素和一个          -->
<!-- Right_Model 元素。-->
<!-- ********************************************************** -->

<!ELEMENT Association ((Left_Model | Right_Model)*) >
<!ATTLIST Association relation CDATA #REQUIRED >


<!-- ********************************************************** -->
<!-- Left_Model 元素定义 Spectrum 关联中的     -->
<!-- 右侧模型。-->
<!--                                                            -->
<!-- Left_Model 元素仅允许包含一个子       -->
<!-- 元素。-->
<!-- ********************************************************** -->

<!ELEMENT Left_Model (Device                        |
                      Port                          |
                      Topology_Container            |
                      Location_Container            |
                      EventModel                    |
                      Model
                     ) >



<!-- ********************************************************** -->
<!-- Right_Model 元素定义 Spectrum 关联中的     -->
<!-- 右侧模型。-->
<!--                                                            -->
<!-- Right_Model 元素仅允许包含一个子       -->
<!-- 元素。-->
<!-- ********************************************************** -->
<!ELEMENT Right_Model (Device                       |
                       Port                         |
                       Topology_Container           |
                       Location_Container           |
                       EventModel                   |
                       Model
                      ) >


<!-- ********************************************************** -->
<!-- Model 元素可用于表示任何 SPECTRUM 模型。-->
<!--                                                            -->
<!-- 必须提供 model_type 和 name 来定义模型。-->
<!-- 应使用“model_type”和“name”来唯一标识  -->
<!-- 模型。但是，如果提供了“model_handle”，则不使用 -->
<!-- “model_type”和“name”的值。-->
<!-- ********************************************************** -->
```

```
<!ELEMENT Model ( #PCDATA ) >
<!ATTLIST Model
          name           CDATA    #REQUIRED
          model_type     CDATA    #REQUIRED
          model_handle   CDATA    #IMPLIED >


<!-- ********************************************************** -->
<!-- 用于自定义视图的 GenericView 元素       -->
<!-- 可包含任意数量的 GenericView_Container 和 Device  -->
<!-- 元素。-->
<!--                                                            -->
<!-- 此元素有 3 个必需属性：containment_relation、 -->
<!-- model_type 和 name。-->
<!-- ********************************************************** -->

<!ELEMENT GenericView ((GenericView_Container | Device )*) >

<!ATTLIST GenericView
        containment_relation  CDATA          #REQUIRED
        model_type            CDATA          #REQUIRED
        name          CDATA                  #REQUIRED
        complete_topology   (false | true)   #IMPLIED >

<!-- ********************************************************** -->
<!-- 用于 GenericView 容器的 GenericView_Container 元素       -->
<!-- 可以包含任意数量的         -->
<!-- GenericView_Container 和 Device 元素。-->
<!--                                                            -->
<!-- 此元素需要 model_type 和 name 属性，         -->
<!-- 不需要 containment_relation 属性。如果没有-->
<!-- 指定该属性，则将继承父项的        -->
<!-- containment_relation。Model_Attr 可用于多行      -->
<!-- 文本字符串 SPECTRUM 属性或列表属性。-->
<!-- ********************************************************** -->

<!ELEMENT GenericView_Container ( GenericView_Container |
        Device
        )*>
<!ATTLIST GenericView_Container
        name                CDATA          #REQUIRED
        model_type          CDATA          #REQUIRED
        containment_relation CDATA         #IMPLIED >

<!-- ********************************************************** -->
<!-- Model_Attr 用于多行文本字符串或列表       -->
<!-- SPECTRUM 属性。-->
<!--                                                            -->
<!-- 此元素需要 attr_id 来指定          -->
<!-- SPECTRUM 属性。此元素可以包含适用于 SPECTRUM 文本字符串    -->
<!-- 属性类型的多行文本字符串或适用于        -->
<!-- SPECTRUM 列表属性的多个 List_Value 元素。-->
<!-- ********************************************************** -->
```

```
<!ELEMENT Model_Attr ( #PCDATA | List_Value )* >
<!ATTLIST Model_Attr
                        attr_id   CDATA          #REQUIRED >

<!-- ********************************************************** -->
<!-- List_Value 用于 SPECTRUM 列表属性值。-->
<!--                                                            -->
<!-- 每个 List_Value 包含一个 PCDATA，用作        -->
<!-- 列表属性的一个实例值。-->
<!-- ********************************************************** -->
<!ELEMENT List_Value ( #PCDATA ) >

<!-- ********************************************************** -->
<!-- 服务级别管理拓扑元素。-->
<!-- ********************************************************** -->

<!ELEMENT CustomerManager ( SM_Customer |
                                SM_CustomerGroup
                                )*>

<!ATTLIST CustomerManager
        name                  CDATA                 #IMPLIED
        containment_relation  ( Groups_Customers )  #IMPLIED
        model_type            ( CustomerManager )   #IMPLIED
        >

<!ELEMENT SM_ServiceMgr ( SM_Service )*>

<!ATTLIST SM_ServiceMgr
        name            CDATA                         #IMPLIED
        containment_relation          ( SlmContains )   #IMPLIED
        model_type      ( SM_ServiceMgr )               #IMPLIED
        >

<!ELEMENT SM_SLA_Mgr ( SM_SLA )*>

<!ATTLIST SM_SLA_Mgr
        name                  CDATA                 #IMPLIED
        containment_relation  ( SlmContainsSLAs )   #IMPLIED
        model_type            ( SM_SLA_Mgr )        #IMPLIED
        >

<!ELEMENT SM_Service_Mgt ( CustomerManager |
                        SM_ServiceMgr   |
                        SM_SLA_Mgr
                        )*>

<!ATTLIST SM_Service_Mgt
        name            CDATA                               #IMPLIED
        containment_relation  ( SlmHasServiceComponent )    #IMPLIED
        model_type            ( SM_Service_Mgt )            #IMPLIED
        >
```

```
<!-- Correlation 元素表示根模型 Correlation_Manager -->
<!-- (0x10469)。它只能包含 Correlation_Domain 元素，且  -->
<!-- 没有属性。所有 Correlation_Domains 将通过 CORRELATES 关系          -->
<!-- 与 Correlation Manager 相关联。-->

<!-- 由于 Correlation_Manager 是一个独特的模型，此元素    -->
<!-- 实际上不会导致 SpectroSERVER 创建模型。它表示     -->
<!-- 一个预先存在的模型。-->

<!ELEMENT Correlation ( Correlation_Domain )*>

<!-- Correlation_Domain 可包含任意数量的 Device、Port、       -->
<!-- Model_Attr 或 GenericView_Container。它们将通过 CORRELATES 关系与      -->
<!-- Correlation_Domain 相关联。Correlation_   -->
<!-- Domains 也没有属性。-->

<!ELEMENT Correlation_Domain (  Device                |
                                Port                  |
                                Model_Attr            |
                                GenericView_Container
                                )*>

<!ATTLIST Correlation_Domain
        name            CDATA                   #REQUIRED
        >

<!ELEMENT RTM_Test ( #PCDATA) >

<!ATTLIST RTM_Test
        name            CDATA             #REQUIRED
        model_type      ( RTM_Test )      #IMPLIED
        >

<!ELEMENT SM_Service ( SM_Service         |
                       SM_AttrMonitor     |
                       SM_LatencyMon      |
                       SM_ConnectMon      |
                       Device             |
                       Port               |
                       Topology_Container |
                       RTM_Test           |
                       MonitorPolicy_Attr |
                       Schedule
                       )*>

<!-- ************************************************************************
-->
<!-- 表示 SM_Service 模型                                                  -->
<!--                                                                        -->
<!-- Criticality 可以是以下值之一                                        -->
<!--                                                                        -->
```

```
<!--  10 - 低                                                              -->
<!--  15 - 中低                                                          -->
<!--  20 - 中                                                              -->
<!--  25 - 中高                                                          -->
<!--  10 - 高                                                              -->
<!--                                                                       -->
<!-- AttrToWatch 可以是以下值之一                                    -->
<!--                                                                       -->
<!--  Condition - 可用于大部分模型，策略 1-5                  -->
<!--  Contact_Status - 通常用于设备模型，策略 10-13     -->
<!--  Port_Status - 用于接口模型，策略 14-17               -->
<!--  LatestErrorStatus - 用于 RTM_Test 模型，策略 18-21          -->
<!--  或 Response_Time                                                     -->
<!--  RM_Condition - SM_AttrMonitor 或 SM_Service 模型，策略 6-9      -->
<!--  或 Service_Health                                                    -->
<!--                                                                       -->
<!-- MonitorPolicy_ID - GlobalConfig 的 SLM_DefaultPolicies 的索引 1-21   -->
<!--                                                                       -->
<!--  1  - 条件-汇总                                                  -->
<!--  2  - 条件-冗余                                                  -->
<!--  3  - 条件-高灵敏度                                            -->
<!--  4  - 条件-低灵敏度                                            -->
<!--  5  - 条件-百分比                                                -->
<!--  6  - 服务运行状况-冗余                                        -->
<!--  7  - 服务运行状况-高灵敏度                                  -->
<!--  8  - 服务运行状况-低灵敏度                                  -->
<!--  9  - 服务运行状况-百分比                                      -->
<!--  10 - 联系状态-冗余                                            -->
<!--  11 - 联系状态-高灵敏度                                      -->
<!--  12 - 联系状态-低灵敏度                                      -->
<!--  13 - 联系状态-百分比                                          -->
<!--  14 - 端口状态-冗余                                            -->
<!--  15 - 端口状态-高灵敏度                                      -->
<!--  16 - 端口状态-低灵敏度                                      -->
<!--  17 - 端口状态-百分比                                          -->
<!--  18 - 响应时间-冗余                                            -->
<!--  19 - 响应时间-高灵敏度                                      -->
<!--  20 - 响应时间-低灵敏度                                      -->
<!--  21 - 响应时间-低百分比                                        -->
<!--                                                                       -->
<!-- ************************************************************************
-->

<!ATTLIST SM_Service
        name                    CDATA                   #REQUIRED
        containment_relation    ( SlmMonitors |
                                  SlmWatchesContainer |
                                  MaintenenceScheduledBy ) #IMPLIED
        Criticality   ( 10 | 15 | 20 | 25 | 30 )        #IMPLIED

        AttrToWatch             CDATA                   #IMPLIED
```

```
        MonitorPolicy_ID        CDATA                   #IMPLIED
        is_managed              (true  | false)         #IMPLIED
        Generate_Service_Alarms (true  | false)         #IMPLIED
        Security_String         CDATA                   #IMPLIED
        model_type        ( SM_Service )                #IMPLIED
        >

<!ELEMENT SM_AttrMonitor ( SM_Service         |
                           SM_AttrMonitor     |
                           SM_LatencyMon      |
                           SM_ConnectMon      |
                           Device             |
                           Port               |
                           Topology_Container |
                           RTM_Test           |
                           MonitorPolicy_Attr |
                           )*>


<!-- ********************************************************************
-->
<!-- 表示 SM_AttrMonitor 模型                                          -->
<!--                                                                    -->
<!-- AttrToWatch 可以是以下值之一                                      -->
<!--                                                                    -->
<!--  Condition - 可用于大部分模型，策略 1-5                        -->
<!--  Contact_Status - 通常用于设备模型，策略 10-13              -->
<!--  Port_Status - 用于接口模型，策略 14-17                        -->
<!--  LatestErrorStatus - 用于 RTM_Test 模型，策略 18-21           -->
<!--  或 Response_Time                                                  -->
<!--  RM_Condition - SM_AttrMonitor 或 SM_Service 模型，策略 6-9      -->
<!--  或 Service_Health                                                 -->
<!--                                                                    -->
<!-- MonitorPolicy_ID - GlobalConfig 的 SLM_DefaultPolicies 的索引 1-21   -->
<!--                                                                    -->
<!--  1  - 条件-汇总                                                 -->
<!--  2  - 条件-冗余                                                 -->
<!--  3  - 条件-高灵敏度                                             -->
<!--  4  - 条件-低灵敏度                                             -->
<!--  5  - 条件-百分比                                               -->
<!--  6  - 服务运行状况-冗余                                         -->
<!--  7  - 服务运行状况-高灵敏度                                     -->
<!--  8  - 服务运行状况-低灵敏度                                     -->
<!--  9  - 服务运行状况-百分比
<!--  10 - 联系状态-冗余                                             -->
<!--  11 - 联系状态-高灵敏度                                         -->
<!--  12 - 联系状态-低灵敏度                                         -->
<!--  13 - 联系状态-百分比                                           -->
<!--  14 - 端口状态-冗余                                             -->
<!--  15 - 端口状态-高灵敏度                                         -->
<!--  16 - 端口状态-低灵敏度                                         -->
```

```
<!--  17 - 端口状态-百分比                                                    -->
<!--  18 - 响应时间-冗余                                                     -->
<!--  19 - 响应时间-高灵敏度                                                 -->
<!--  20 - 响应时间-低灵敏度                                                 -->
<!--  21 - 响应时间-低百分比                                                 -->
<!--                                                                        -->
<!--  Special_Cause_List - 可用于指定        -->
<!--                       包括或排除影响         -->
<!--                       服务或资源监控模型的服务运行状况       -->
<!--                       的警报原因的列表或范围。仅适用于        -->
<!--                       条件为 AttrToWatch 时。-->
<!--                                                                        -->
<!--  Cause_List_Control - 指定 Special_Cause_List 的使用方法。-->
<!--                       0 - 未使用                                       -->
<!--                       1 - 包括                                        -->
<!--                       2 - 排除                                        -->
<!--                                                                        -->
<!-- ************************************************************************
-->


<!ATTLIST SM_AttrMonitor
        name                    CDATA                       #REQUIRED
        containment_relation    ( SlmMonitors |
                                  SlmWatchesContainer )  #IMPLIED
        is_managed              (true  | false)           #IMPLIED
        AttrToWatch             CDATA                       #IMPLIED
        MonitorPolicy_ID        CDATA                       #IMPLIED
        Generate_Service_Alarms (true  | false)           #IMPLIED
        model_type      ( SM_AttrMonitor )                  #IMPLIED
        >

<!ELEMENT MonitorPolicy_Attr ( #PCDATA ) >

<!ELEMENT SM_SLA ( SM_Service    |
                SM_Guarantee |
                Schedule
                )*>

<!-- ************************************************************************
-->
<!-- 表示 SM_Guarantee 模型                                                 -->
<!--                                                                        -->
<!-- SLAControl 可具有以下值                          -->
<!--                                                                        -->
<!--  0 = 非活动                                                           -->
<!--  1 = 活动                                                             -->
<!--                                                                        -->
<!-- ************************************************************************
-->
```

```
<!ATTLIST SM_SLA
        name              CDATA                   #REQUIRED
        containment_relation  ( SlmHasGuarantee |
                                SlmGuarantees |
                                SlaPeriod )       #IMPLIED
        is_managed          (true  | false)       #IMPLIED
        SLA_Control         ( 0 | 1 )             #IMPLIED
        SLA_ExpirationDate    CDATA               #IMPLIED
        SLA_Notes             CDATA               #IMPLIED
        SLA_Description       CDATA               #IMPLIED
        Security_String       CDATA               #IMPLIED
        model_type          ( SM_SLA )            #IMPLIED
        >

<!ELEMENT SM_Guarantee ( SM_Service      |
                         SM_AttrMonitor  |
                         SM_LatencyMon   |
                         SM_ConnectMon   |
                         Schedule
                         )*>

<!-- ************************************************************************
-->
<!-- 表示 SM_Guarantee 模型                                                  -->
<!--                                                                         -->
<!-- GuaranteeControl 可具有以下值                                           -->
<!--                                                                         -->
<!--  0 = 非活动                                                             -->
<!--  1 = 活动                                                               -->
<!--                                                                         -->
<!-- GuranteeType 可具有以下值                                               -->
<!--                                                                         -->
<!--  0 = 可用性                                                             -->
<!--  1 = 性能                                                               -->
<!--  2 = 平均修复时间                                                       -->
<!--  3 = 最大停机时间                                                       -->
<!--                                                                         -->
<!-- ServiceHealthType 可具有以下值                                          -->
<!--                                                                         -->
<!--                       1 - 宕机                                          -->
<!--                       2 - 已降级                                        -->
<!--                                                                         -->
<!-- ************************************************************************
-->

<!ATTLIST SM_Guarantee
        name                        CDATA          #REQUIRED
        containment_relation  ( SlmIsMeasuredBy |
                                SlmSchedulesGuarantee )  #IMPLIED
        is_managed            (true  | false)      #IMPLIED
        GuaranteeControl      ( 0 | 1 )            #IMPLIED
        GuaranteeType         ( 0 | 1 | 2 | 3 )    #REQUIRED
        ServiceHealthType     ( 1 | 2 )            #IMPLIED
```

```
        WarningThreshold            CDATA          #IMPLIED
        WarningThresholdPercent     CDATA          #IMPLIED
        ViolationThreshold          CDATA          #IMPLIED
        ViolationThresholdPercent   CDATA          #IMPLIED
        GuaranteeNotes              CDATA          #IMPLIED
        GuaranteeDescription        CDATA          #IMPLIED
        model_type             ( SM_Guarantee )    #IMPLIED
        MOT_Threshold               CDATA          #IMPLIED
        MTTR_Threshold              CDATA          #IMPLIED
        MTBF_Threshold              CDATA          #IMPLIED
        >

<!ELEMENT SM_LatencyMon ( Topology_Container |
                          MonitorPolicy_Attr )*>

<!ATTLIST SM_LatencyMon
        name                    CDATA                #REQUIRED
        containment_relation  ( SlmMonitors |
                                SlmWatchesContainer ) #IMPLIED
        is_managed            ( true  | false )      #IMPLIED
        DefaultMaxRTT           CDATA                #IMPLIED
        DefaultMeasureInterval  CDATA                #IMPLIED
        model_type            ( SM_LatencyMon )      #IMPLIED
        >

<!ELEMENT SM_CustomerGroup ( SM_CustomerGroup |
                      SM_Customer
                      )*>

<!ATTLIST SM_CustomerGroup
        name                    CDATA                #REQUIRED
        containment_relation ( Groups_Customers )    #IMPLIED
        model_type           ( SM_CustomerGroup )    #IMPLIED
        >

<!ELEMENT SM_Customer ( SM_Service |
                        SM_SLA     |
                      )*>

<!-- ********************************************************************
-->
<!-- 表示 SM_Customer 模型                                          -->
<!--                                                                    -->
<!-- Criticality 可以是以下值之一                                   -->
<!--                                                                    -->
<!--  10 - 低                                                           -->
<!--  15 - 中低                                                      -->
<!--  20 - 中                                                          -->
<!--  25 - 中高                                                     -->
<!--  10 - 高                                                           -->
<!--                                                                    -->
<!-- ********************************************************************
-->
```

```
<!ATTLIST SM_Customer
        name                            CDATA  #REQUIRED
        containment_relation  ( SlmAgreesTo |
                                SlmUses )     #IMPLIED
        Security_String                 CDATA  #IMPLIED
        CustomerID                      CDATA  #IMPLIED
        Criticality                     CDATA  #IMPLIED
        CustomerField4                  CDATA  #IMPLIED
        CustomerField5                  CDATA  #IMPLIED
        CustomerField6                  CDATA  #IMPLIED
        CustomerField7                  CDATA  #IMPLIED
        Contact_Name                    CDATA  #IMPLIED
        Contact_Title                   CDATA  #IMPLIED
        Contact_Location                CDATA  #IMPLIED
        Email_Address                   CDATA  #IMPLIED
        Phone_Number                    CDATA  #IMPLIED
        Mobile_Phone_Number             CDATA  #IMPLIED
        Pager_Number                    CDATA  #IMPLIED
        Fax_Number                      CDATA  #IMPLIED
        User_Defined_1                  CDATA  #IMPLIED
        User_Defined_2                  CDATA  #IMPLIED
        User_Defined_3                  CDATA  #IMPLIED
        User_Defined_4                  CDATA  #IMPLIED
        Secondary_Contact_Name          CDATA  #IMPLIED
        Secondary_Contact_Title         CDATA  #IMPLIED
        Secondary_Contact_Location      CDATA  #IMPLIED
        Secondary_Email_Address         CDATA  #IMPLIED
        Secondary_Phone_Number          CDATA  #IMPLIED
        Secondary_Mobile_Phone_Number   CDATA  #IMPLIED
        Secondary_Pager_Number          CDATA  #IMPLIED
        Secondary_Fax_Number            CDATA  #IMPLIED
        Secondary_User_Defined_1        CDATA  #IMPLIED
        Secondary_User_Defined_2        CDATA  #IMPLIED
        Secondary_User_Defined_3        CDATA  #IMPLIED
        Secondary_User_Defined_4        CDATA  #IMPLIED
        model_type            ( SM_Customer )  #IMPLIED
        >

<!-- ************************************************************************
-->
<!-- 表示 GlobalCollection 模型                                              -->
<!--                                                                        -->
<!-- ************************************************************************
-->
<!ELEMENT GlobalCollection (( Device             |
                        Topology_Container |
                        Location_Container )*) >
```

```
<!ATTLIST GlobalCollection
        name              CDATA                 #REQUIRED
        containment_relation  CDATA             "GlobalCollect"
        collectionDescription CDATA             #IMPLIED
        Security_String       CDATA             #IMPLIED >
```

附录 C： XML 示例

本部分包含 XML 示例，可帮助您使用 DTD。

注意：每个示例中的元素名称以粗体突出显示。将名称设置为粗体可提高示例易读性；其并不是暗示 XML 输入文件所需的格式设置。

示例 1：导入到拓扑视图

此示例显示将信息导入 CA Spectrum 拓扑视图的基本输入文件。此文件在拓扑视图中创建一个网络容器模型。在该网络容器中，创建一个 LAN 容器模型。在该 LAN 容器中，创建两台设备。用 DNS 名称死锁标识一台设备，用 IP 地址标识另一台设备。

Topology 元素的 complete_topology 属性设置为 False。在这种情况下，CA Spectrum 考虑先前存在于拓扑视图中的其他模型。因此，此文件仅旨在为 XML 文件中列出但尚未建模的条目创建模型。创建的模型放置在文件中指定的拓扑层次结构中。先前存在于拓扑层次结构中的模型不会被重新发现，但会移动到在文件中指定的容器。

注意：当 complete_topology 设置为 false 时，容器中未列入导入文件的现有模型*不会*发送至“Lost and Found”。当 complete_topology 设置为 true 时，这些模型*会*发送至“Lost and Found”。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- ********************************** -->
<!-- 此部分用于拓扑视图导入 -->
<!-- ********************************** -->
<Topology complete_topology="false">
        <Device ip_dnsname="10.253.9.17" model_type="GnSNMPDev"
                community_string="public"/>
        <Device ip_dnsname="nmcss52-5" />
        <Topology_Container model_type="Network" name="My Network"
                Security_String="public" subnet_address="10.253.0.0"
                subnet_mask="255.255.0.0">
```

```
                <Topology_Container model_type="Lan" name="Lan1"
                        Security_String="public" subnet_address="10.253.9.0"
                        subnet_mask="255.255.255.0">
                        <Device ip_dnsname="deadlock" />
                        <Device ip_dnsname="10.253.9.18" poll_interval="333"
/>
                </Topology_Container>
        </Topology_Container>
</Topology>
</Import>
```

示例 2：创建连接

下例显示的是在两个 ATM 电路之间创建的连接以及在两个帧中继电路之间创建的连接。此示例还显示 FrameRelay DLCI 端口和 ATM VCL 端口之间的连接。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<Topology complete_topology="false">
        <Connection create_pipe="false">
                <Device ip_dnsname="10.253.32.225">
                        <Port identifier_name="atmVclTableInstance"
                                identifier_value="5.0.5"
                                circuit_name="ATM Link1"
                                circuit_id = "ATM 5017" />
                </Device>
                <Device ip_dnsname="192.168.52.25">
                        <Port identifier_name="atmVclTableInstance"
                                identifier_value="3.0.12"
                                circuit_name="ATM Link1"
                                circuit_id = "ATM 5017" />
                </Device>
        </Connection>
        <Connection>
                <Device ip_dnsname="10.253.9.18">
                        <Port identifier_name="frCircuitTableInstance"
                                identifier_value="2.27"/>
                </Device>
                <Device ip_dnsname="nmcss52-5">
                        <Port identifier_name="frCircuitTableInstance
                                identifier_value="4.161"/>
                </Device>
        </Connection>
```

```
        <!-- ****************************************** -->
        <!-- FrameRelay DLCI 端口和 -->
        <!-- ATM VCL 端口之间的连接。 -->
        <!-- ****************************************** -->
        <Connection>
                <Device ip_dnsname="10.253.9.18">
                        <Port identifier_name="frCircuitTableInstance"
                                identifier_value="2.27"/>
                </Device>
                <Device ip_dnsname="10.253.32.225">
                        <Port identifier_name="atmVclTableInstance"
                                identifier_value="5.0.17"/>
                </Device>
        </Connection>
        <Connection>
                <Device ip_dnsname="nmcss52-5">
                        <Port identifier_name="ifIndex" identifier_value="3"/>
                </Device>
                <Device ip_dnsname="10.253.9.17">
                        <Port identifier_name="ifPhysAddress"
                                identifier_value="0:4:27:C:91:C0"/>
                </Device>
        </Connection>
<Topology>
</Import>
```

示例 3：更新和销毁

此示例说明 Update 和 Destroy 元素的用法。

Update 元素包含 Location_Container 元素。此示例使用 Location_Container 元素的 name 和 model_name 属性更新模型名称。name 属性设置为等于当前名称并标识要更新的模型。model_name 属性将 name 属性的值更新为 Peace2。

Update 元素还包含 Device 元素和 Port 元素。identifer_name 和 identifier_value 属性用于标识要更新的端口。其他指定的属性为其值已更新的属性。端口模型名称将更改为 port 2，poll_status 将更改为 False。

Destroy 元素将消除设备模型死锁。与死锁关联的任何连接或端口将自动销毁。Building 容器模型 Durham 也会被消除。包含在 Building 容器 Durham 中的任何模型都将发送至“Lost and Found”。

Destroy 元素还删除设备 nmcss52-5 上的指定端口和 nmcss52-3 上的指定端口之间的连接。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
```

```
<Import>

<!-- ********************************************** -->
<!-- 模块更新.....................................-->
<!-- ********************************************** -->
        <Update>
<!-- ********************************************** -->
<!-- 将容器 Peace 的模型名称从 Peace 更改为-->
<!-- Peace2 ..................................... ....-->
<!-- ********************************************** -->
                <Location_Container model_type="Building" name="Peace"
                        model_name="Peace2"/>
<!-- ********************************************* -->
<!-- 在设备 nmcss52-5 上更新端口 ifIndex=2   -->
<!-- ********************************************* -->
                <Device ip_dnsname="nmcss52-5">
                        <Port identifier_name="ifIndex" identifier_value="2"
                                model_name="port 2" poll_status="false" />
                </Device>
        </Update>
  <!-- ****************************** -->
  <!-- 销毁模型和连接。-->
  <!-- ****************************** -->
        <Destroy>
                <Device ip_dnsname="deadlock"/>
                <Location_Container  model_type="Building" name="Durham" />
                <Connection>
                        <Device ip_dnsname="nmcss52-5">
                                <Port identifier_name="ifIndex"
                                        identifier_value="1"/>
                        </Device>
                        <Device ip_dnsname="10.253.9.17">
                                <Port identifier_name="ipAddress"
                                        identifier_value="10.253.8.18"/>
                        </Device>
                </Connection>
        </Destroy>
</Import>
```

示例 4：创建、更新和销毁

以下 XML 文件说明包含在 DTD 中的元素的大部分功能。此文件在拓扑视图和位置视图中创建数据、创建连接、更新属性、销毁模型和连接。

此 XML 文件的第一部分在拓扑视图中创建模型并创建这些模型之间的连接。这一部分以 Topology 元素开头 (< Topology...>)。先创建容器模型和设备模型，然后建立连接。当 Topology 元素结束 (</Topology>) 时，这一部分即结束。

Topology 元素关闭后，将在另一个部分中创建另一个连接。此部分说明，您无需在 Topology 元素中嵌套 Connection 元素即可创建连接。

此文件的下一部分以 Location 元素开头 (< Location...>)。这一部分演示在位置视图中创建容器和设备模型。当 Location 元素结束 (</Location>) 时，这一部分即结束。

下一部分以 Update 元素开头 (<Update>)。在这一部分中，将修改容器、设备和端口的属性值。通过查看 XML 文件，您无法知道所表示的每个元素的当前属性值。因此，不容易识别正在更新的元素。通常，每个元素都包含一个唯一标识要更新的模型或端口的属性。其余属性指定为更新其值。例如，第一个元素是 Location_Container 元素。name 属性唯一标识模型。model_type 属性指定为更新其值，也许从 Region 更新为 Building。唯一可在 Update 元素中定义的层次结构是指定要更新的端口的 Device/Port 层次结构。当 Update 元素结束 (</Update>) 时，这一部分即结束。

此文件的最后一个部分使用 Destroy 元素。Destroy 元素将消除：

- IP 地址为 10.253.9.19 的设备
- 容器模型 Durham
- 设备 nmcss52-5 上的指定端口和地址为 10.253.9.17 的设备上的指定端口之间的连接

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- ********************************** -->
<!-- 此部分用于拓扑视图导入 -->
<!-- ********************************** -->
        <Topology discover_connections="false" complete_topology="false">
                <Device ip_dnsname="10.253.9.109" model_type="GnSNMPDev"
                        community_string="public" is_managed="false"/>
                <Device ip_dnsname="10.253.9.17"
                        poll_interval="333" log_ratio="11"/>
                <Device ip_dnsname="10.253.9.19" community_string="public"/>
                <Device ip_dnsname="nmcss52-5" />
```

```
<Topology_Container model_type="Network" name="My Network"
        Security_String="public" subnet_address="10.253.0.0"
        subnet_mask="255.255.0.0" complete_topology="true">
        <Topology_Container model_type="Lan"

                name="MyLan" Security_String="public"

                subnet_address="10.253.9.0"
                subnet_mask="255.255.255.0">
                <Device ip_dnsname="10.253.9.18"
                        community_string="public"
                        poll_interval="333"
                        log_ratio="5"/>
        </Topology_Container>
</Topology_Container>
<Topology_Container model_type="IPClassC" name="my_net"
        subnet_address="172.19.57.0">
        <Device  model_type="Pingable"
                ip_dnsname="172.19.57.91"/>
        <Device  model_type="Fanout" ip_dnsname="1.2.3.4"/>
        <Device  ip_dnsname="10.253.9.16"
                community_string="public"/>
</Topology_Container>
<Topology_Container model_type="Lan" name="lan2"
        Security_String="public" subnet_address="10.253.7.0"
        subnet_mask="255.255.255.0" complete_topology="true">
        <Device ip_dnsname="10.253.7.17"
                community_string="public"
                poll_interval="333" log_ratio="11"/>
        <Device ip_dnsname="10.253.32.101"/>
        <Device ip_dnsname="192.168.125.161"
                model_type="GnSNMPDev"/>
</Topology_Container>
<Device ip_dnsname="172.19.57.92" />
<Device ip_dnsname="172.19.57.93" />
<Device ip_dnsname="10.253.32.225" model_type="M46_04"/>
<Connection>
        <Device ip_dnsname="172.19.57.93">
                <Port identifier_name=
"frCircuitTableInstance"
                        identifier_value="4.161"/>
        </Device>
        <Device ip_dnsname="192.168.125.161">
                <Port identifier_name
="frCircuitTableInstance"
                        identifier_value="2.161"/>
        </Device>
</Connection>
```

```
                <Connection create_pipe="false">
                        <Device ip_dnsname="10.253.32.101">
                                <Port identifier_name ="atmVclTableInstance"
                                        identifier_value="3.1.52"/>
                        </Device>
                        <Device ip_dnsname="10.253.32.225">
                                <Port identifier_name ="atmVclTableInstance"
                                        identifier_value="5.0.68"
                                        circuit_name="ATM 68"
                                        circuit_id ="ATM ID 68"/>
                        </Device>
                </Connection>
                <Connection>
                        <Device ip_dnsname="nmcss52-5">
                                <Port identifier_name="ifIndex"
                                        identifier_value="1"/>
                        </Device>
                        <Device ip_dnsname="10.253.9.17">
                                <Port identifier_name="ipAddress"
                                        identifier_value="10.253.8.18"/>
                        </Device>
                </Connection>
        </Topology>
        <Connection>
                <Device ip_dnsname="172.19.57.92">
                        <Port identifier_name="ifPhysAddress"
                                identifier_value="0:E0:63:7C:19:61"/>
                </Device>
                <Device ip_dnsname="10.253.9.17">
                        <Port identifier_name="ipAddress"
                                identifier_value="10.253.8.65"/>
                </Device>
        </Connection>
<!-- ********************************** -->
<!-- 此部分用于位置视图导入 -->
<!-- ********************************** -->
        <Location complete_topology="true">
                <Location_Container model_type="Country" name="USA"
                        Security_String="whatever">
                        <Location_Container model_type="Region"
                                name="New Hampshire"
                                complete_topology="false">
                                <Location_Container model_type="Site"
                                name="Durham"
                                        <Device ip_dnsname = "10.253.32.10"/>
                                        <Device ip_dnsname = "172.19.57.93" />
                                </Location_Container>
                        </Location_Container>
                </Location_Container>
```

```
                <Location_Container model_type="Building" name="Durham"
                        Security_String="public">
                        <Location_Container model_type="Room" name="my_room"
                                Security_String="hahaha">
                                <Device ip_dnsname="10.253.9.16"
                                        community_string="public"/>
                                <Device ip_dnsname="10.253.9.17" />
                                <Device ip_dnsname = "10.253.9.18"/>
                        </Location_Container>
                </Location_Container>
                <Location_Container model_type="Building" name="Peace"
                        Security_String="aprisma">
                        <Location_Container model_type="Room" name= "Lab 1">
                                <Device ip_dnsname="10.253.7.17"
                                        community_string="public"/>
                                <Device ip_dnsname="192.168.125.161"/>
                        </Location_Container>
                </Location_Container>
        </Location>
<!-- ************************************** -->
<!-- 此部分用于模型更新                      -->
<!-- ************************************** -->
        <Update>
                <Location_Container model_type="Building" name="Peace"
                        model_modify_author="ltang"/>
                <Device ip_dnsname="172.19.57.93" poll_interval="101"
                        model_name="haha" />
    <!-- ************************************** -->
    <!-- 此部分在设备 nmcss52-5 上 -->
    <!-- 更新端口 ifIndex=2         -->
    <!-- ************************************** -->
                <Device ip_dnsname="nmcss52-5">
                        <Port identifier_name="ifIndex" identifier_value="2"
                                model_name="port 2" poll_interval="1103"
                                poll_status="false" log_ratio="12"/>
                </Device>
                <Topology_Container model_type="Lan" name="lan2"
                        Security_String="top secret"/>
        </Update>
```

```
<!-- ******************************************** -->
<!-- 此部分用于删除模型和连接 -->
<!-- ******************************************** -->
        <Destroy>
                <Device ip_dnsname="10.253.9.19"/>
                <Location_Container model_type="Building" name="Durham"/>
                <Connection>
                        <Device ip_dnsname="nmcss52-5">
                                <Port identifier_name="ifIndex"
                                        identifier_value="1"/>
                        </Device>
                        <Device ip_dnsname="10.253.9.17">
                                <Port identifier_name="ipAddress"
                                        identifier_value="10.253.8.18"/>
                        </Device>
                </Connection>
        </Destroy>
</Import>
```

附录 D： .modelinggatewayresource.xml

此部分包含 .modelinggatewayresource.xml 文件的副本。但是，此副本可能不是此文件的最新版本。有关最新版本，请使用您的建模网关工具包随附的实际文件。

```
<?xml version="1.0" standalone="no"?>

<TopologyImportExportResourceFile>

<!-- ************************************** -->
<!-- 用于拓扑导出和导入的 SPECTRUM      -->
<!-- 属性名称和 ID。-->
<!-- ************************************** -->

<Attributes
    circuit_id                  = "0xc4042f"
    circuit_name                = "0xc40430"
    community_string            = "0x10024"
    agent_port                  = "0x10023"
    DeviceType                  = "0x23000e"
    is_managed                  = "0x1295d"
    log_ratio                   = "0x10072"
    manager_name                = "0x3dc0009"
    model_modify_author         = "0x11025"
    model_name                  = "0x1006e"
    name                        = "0x1006e"
    poll_interval               = "0x10071"
    poll_status                 = "0x1154f"
    TryCount                    = "0x110c5"
    Security_String             = "0x10009"
    subnet_address              = "0x1027f"
    subnet_mask                 = "0x110b8"
    subnet_list                 = "0x11953"
    TimeOut                     = "0x110c4"
    trapIPAddress               = "0x3dc0007"
    unique_id                   = "0x3dc0004"
    Value_When_Orange           = "0x1000d"
    Value_When_Red              = "0x1000e"
    Value_When_Yellow           = "0x1000c"

    LatestErrorStatus           = "456008c"
    Response_Time               = "456008c"

    AttrToWatch                 = "0x12a43"
    MonitorPolicy               = "0x12a3e"
```

```
MonitorPolicy_ID              = "0x12a51"
Generate_Service_Alarms       = "0x12a66"
Special_Cause_List            = "0x12b47"
Cause_List_Control            = "0x12d50"

Contact_Status                = "0x10004"
Port_Status                   = "0x10f1b"
RM_Condition                  = "0x12a40"
Service_Health                = "0x12a40"
Criticality                   = "0x1290c"
Condition                     = "0x1000a"
Condition_Value               = "0x1000b"

AccumulationMethod            = "0x4500007"
GuaranteeControl              = "0x4500022"
GuaranteeNotes                = "0x4500021"
GuaranteeDescription          = "0x12a4b"
GuaranteeType                 = "0x4500018"
ServiceHealthType             = "0x4500019"
ViolationThreshold            = "0x450001e"
ViolationThresholdPercent     = "0x4500024"
WarningThreshold              = "0x450001d"
WarningThresholdPercent       = "0x4500023"

SCHED_Daily_Repeat_Limit      = "0x1299a"
SCHED_Duration                = "0x12993"
SCHED_Recurrence_Multiplier   = "0x1299b"
SCHED_Recurrence              = "0x12994"
SCHED_Start_DoM               = "0x12991"
SCHED_Start_DoW               = "0x12990"
SCHED_Start_Hour              = "0x1298f"
SCHED_Start_Minute            = "0x1298e"
SCHED_Start_Month             = "0x12992"
SCHED_Start_Day               = "0x129e4"
SCHED_DayBitMask              = "0x129da"
SCHED_Start_Year              = "0x129e3"
SCHED_Start_MoY               = "0x12b48"
SCHED_Description             = "0x12bbc"

SLA_Control                   = "0x4500015"
SLA_Notes                     = "0x4500017"
SLA_ExpirationDate            = "0x4500025"
SLA_Description               = "0x12a4b"

DefaultMaxRTT                 = "0x4500001"
DefaultMeasureInterval        = "0x4500002"

CustomerID                    = "0x12a44"
CustomerField4                = "0x12a39"
CustomerField5                = "0x12a3a"
CustomerField6                = "0x12a3b"
CustomerField7                = "0x12a3c"
```

```
    Contact_Name                  = "0x12a20"
    Contact_Title                 = "0x12a21"
    Contact_Location              = "0x12a22"
    Email_Address                 = "0x12a27"
    Phone_Number                  = "0x12a23"
    Mobile_Phone_Number           = "0x12a24"
    Pager_Number                  = "0x12a25"
    Fax_Number                    = "0x12a26"
    User_Defined_1                = "0x12a28"
    User_Defined_2                = "0x12a29"
    User_Defined_3                = "0x12a2a"
    User_Defined_4                = "0x12a2b"

    Secondary_Contact_Name        = "0x12a2c"
    Secondary_Contact_Title       = "0x12a2d"
    Secondary_Contact_Location    = "0x12a2e"
    Secondary_Email_Address       = "0x12a33"
    Secondary_Phone_Number        = "0x12a2f"
    Secondary_Mobile_Phone_Number = "0x12a30"
    Secondary_Pager_Number        = "0x12a31"
    Secondary_Fax_Number          = "0x12a32"
    Secondary_User_Defined_1      = "0x12a34"
    Secondary_User_Defined_2      = "0x12a35"
    Secondary_User_Defined_3      = "0x12a36"
    Secondary_User_Defined_4      = "0x12a37"

    MOT_Threhold                 = "0x450002c"
    MTBF_Threshold                = "0x4500032"
    MTTR_Thresshold                = "0x450002f"

    Policy_Name_List              = "0x12a4a"
    collectionDescription         = "0x12a67"

/>

<!-- ****************************************** -->
<!-- 可在拓扑导入 XML 文件中使用的 SPECTRUM  -->
<!-- 模型类型名称和句柄。-->
<!-- ****************************************** -->

<ModelTypes
    Universe                      = "0x10091"
    Network                       = "0x1002e"
    Lan                           = "0x1002d"
    IPClassA                      = "0x103d5"
    IPClassB                      = "0x103d6"
    IPClassC                      = "0x103d7"
    LAN_802_3                     = "0x1003c"
    LAN_802_5                     = "0x1003d"
    ATM_NETWORK                   = "0xaa000f"
    EventAdmin                    = "0x3dc0000"
    GlobalCollection              = "0x10474"
```

```
    World                       = "0x10040"
    Country                     = "0x10041"
    Region                      = "0x10042"
    Site                        = "0x10043"
    Sector                      = "0x10044"
    Building                    = "0x10045"
    Section                     = "0x10046"
    Floor                       = "0x10047"
    Room                        = "0x10048"

    Top_Org                     = "0x102cf"
    Enterprise                  = "0x102d0"
    Subsidiary                  = "0x102d1"
    Division                    = "0x102d2"
    Department                  = "0x102d3"
    Org_Section                 = "0x102d4"
    Work_Group                  = "0x102d5"
    Org_Owns                    = "0x102da"

    Schedule                    = "0x10456"
    GnSNMPDev                   = "0x3d0002"
    Fanout                      = "0x100ae"
    Pingable                    = "0x10290"
    WA_Link                     = "0x102e2"
    Unplaced                    = "0x103d8"

    RTM_Test                    = "0x4560000"
    SM_Service                  = "0x1046f"
    SM_AttrMonitor              = "0x1046e"
    SM_LatencyMon               = "0x4500001"
    SM_SLA                      = "0x4500002"
    SM_Guarantee                = "0x4500003"
    SM_Customer                 = "0x1046c"
    SM_CustomerGroup            = "0x10477"
    SM_ConnectMon               = "0x4500000"
    SM_ServiceMgr               = "0x4500006"
    CustomerManager             = "0x10478"
    SM_Service_Mgt              = "0x4500007"
    SM_SLA_Mgr                  = "0x4500008"
    Correlation_Domain          = "0x10467"
    Correlation_Manager         = "0x10469"

/>

<Relations
    Collects                    = "0x10002"
    MaintenenceScheduledBy      = "0x10034"
    SlmAgreesTo                 = "0x4500000"
    SlmGuarantees               = "0x4500001"
    SlmHasGuarantee             = "0x4500002"
    SlmIsMeasuredBy             = "0x4500003"
    SlmMonitors                 = "0x4500004"
```

```
    SlmOwns                         = "0x4500005"
    SlmUses                         = "0x4500006"
    SlmWatchesContainer             = "0x4500007"
    SlmContains                     = "0x4500008"
    SlaPeriod                       = "0x4500009"
    SlmSchedulesGuarantee           = "0x450000c"
    SlmHasServiceComponent          = "0x450000a"
    SlmContainsSLAs                 = "0x450000b"
    Groups_Customers                = "0x1003e"
    GlobalCollect                   = "0x1003b"

/>

<!-- 当 do_not_process_pre_existing_devices_under_container_node    -->
<!-- 设置为 true 时，如果发现容器元素下的设备    -->
<!-- 已存在于 Spectrum，则建模网关将不处理 -->
<!-- 该设备，如更新属性、创建连接等。-->

<ImportConfiguration
    do_not_process_pre_existing_devices_under_container_node = "false"
    import_to_primary_ss_only = "false"
    max_device_creation_threads = "50"
/>

<!-- ********************************************************* -->
<!--                                                           -->
<!-- 这是建模网关导出配置。-->
<!--                                                           -->
<!-- ExportConfiguration 是控制    -->
<!-- 导出内容的配置。-->
<!--                                                           -->
<!-- export_devices：是否导出设备模型               -->
<!--                                                           -->
<!-- export_containers：是否导出容器模型         -->
<!--                                                           -->
<!-- export_port_attributes：是否导出端口属性     -->
<!--                                                           -->
<!-- export_links：是否导出设备链路                   -->
<!--                                                           -->
<!-- export_topology_layout：是否导出设备和容器     -->
<!--                          x,y 坐标            -->
<!--                                                           -->
<!-- export_annotation：是否导出批注             -->
<!--                                                           -->
<!-- export_WA_Link_models：是否导出 WA_Link 模型。-->
<!--                        如果不导出，则 WA_Link 模型将   -->
<!--                        被视为是透明的。两个设备之间   -->
<!--                        通过      -->
<!--                        WA_Link 连接的链路将作为      -->
<!--                        直接链路导出。-->
<!--                                                           -->
```

```
<!-- export_spectrum_settings：是否导出 SPECTRUM 设置-->
<!--                           如，故障     -->
<!--                           隔离、自动发现、       -->
<!--                           VNM 控制等设置...             -->
<!--                                                          -->
<!-- export_user_models：导出 SPECTRUM 用户建模、         -->
<!--                     用户许可、权限等    -->
<!--                                                          -->
<!-- export_service_modeling：导出 SPECTRUM 服务建模-->
<!--                                                          -->
<!-- export_schedules：导出 SPECTRUM 排定。-->
<!--                                                          -->
<!-- export_discovery_configs：导出自动发现的         -->
<!--                           配置。-->
<!--                                                          -->
<!-- ******************************************************* -->

<ExportConfiguration
    export_devices           = "true"
    export_containers        = "true"
    export_port_attributes   = "true"
    export_links             = "true"
    export_topology_layout   = "true"
    export_annotation        = "true"
    export_WA_Link_models    = "true"
    export_spectrum_settings = "true"
    export_user_models       = "true"
    export_service_modeling  = "true"
    export_schedules         = "true"
    export_global_collections = "true"
    export_discovery_configs = "true"
    export_from_primary_ss_only = "false"
    export_policy_manager = "true"
/>

<!-- ******************************************************* -->
<!--                                                          -->
<!-- RootContainerToExport 指定    -->
<!-- SPECTRUM 中要导出的根容器。-->
<!--                                                          -->
<!-- ******************************************************* -->

<RootContainerToExport model_type="Universe" model_name=""/>

<!-- ******************************************************* -->
<!--                                                          -->
<!-- DeviceExportAttributes 是要导出的设备属性 -->
<!-- 的列表。如果上文没有指定属性 ID，  -->
<!-- 则必须分配“attribute_id”。-->
<!--                                                          -->
<!-- ******************************************************* -->
```

```
<DeviceExportAttributes>

    <name/>
    <model_type attribute_id="0x10000"/>
    <community_string/>
    <agent_port/>
    <poll_interval/>
    <is_managed/>
    <poll_status/>
    <Security_String/>
    <TimeOut/>
    <TryCount/>
    <Criticality/>
    <Value_When_Orange/>
    <Value_When_Red/>
    <Value_When_Yellow/>
    <Redundancy_Admin_Status attribute_id="0x11d2c" />
    <Auto_Reconfigure_Interfaces attribute_id="0x11dd4" />
    <Discover_Connection_After_Linkup_Trap attribute_id="0x11d25" />
    <Device_Discovery_After_Reconfiguration attribute_id="0x11d27" />
    <Generate_Redundancy_Alarms attribute_id="0x11dd6" />
    <Create_Sub_Interfaces attribute_id="0x11f3c" />
    <Topology_Relocate_Model attribute_id="0x11a80" />
    <Disable_Trap_Events attribute_id="0x11cd0" />
    <Enable_Spectrum_Management attribute_id="0x1295d" />
    <Hibernate_Device attribute_id="0x12aca" />
    <Enable_Event_Creation attribute_id="0x129f8" />
    <Redundancy_Admin_Status attribute_id="0x11d2c" />
    <DeviceCPUUtilization_Threshold attribute_id="0x12ab9" />
    <DeviceCPUUtilization_Reset attribute_id="0x12abb" />
    <DeviceCPUUtilization_Duration attribute_id="0x12bce" />
    <DeviceMemoryUtilization_Threshold attribute_id="0x12aba" />
    <DeviceMemoryUtilization_Reset attribute_id="0x12abc" />
    <DeviceMemoryUtilization_Duration attribute_id="0x12bcf" />

</DeviceExportAttributes>

<!-- ********************************************************** -->
<!--                                                            -->
<!-- ContainerExportAttributes 是要导出的   -->
<!-- 容器属性。如果上文没有指定属性 ID，  -->
<!-- 则必须分配“attribute_id”。-->
<!--                                                            -->
<!-- ********************************************************** -->
```

```
<ContainerExportAttributes>
    <name/>
    <Security_String/>
    <subnet_address/>
    <subnet_mask/>
    <subnet_list/>
    <Value_When_Orange/>
    <Value_When_Red/>
    <Value_When_Yellow/>
    <SelectMP_port attribute_id="0x118e4" />
 </ContainerExportAttributes>

<!-- ******************************************************** -->
<!--                                                          -->
<!-- PortExportAttributes 是要导出的        -->
<!-- 端口属性。如果上文没有指定属性 ID，  -->
<!-- 则必须分配“attribute_id”。-->
<!--                                                          -->
<!-- 当 export_changed_attribute_only = "true" 时，      -->
<!-- 仅导出其值不等于默认值的    -->
<!-- 端口属性。否则，将导出所有指定的端口    -->
<!-- 属性。-->
<!--                                                          -->
<!-- ******************************************************** -->

<PortExportAttributes export_changed_attribute_only="true" >
    <poll_interval/>
    <poll_status/>
    <ok_to_poll attribute_id="0x11dd8" />
    <PollPortStatus attribute_id="0x1280a" />
    <LockConnection attribute_id="0x129f1" />
    <TimeOut/>
    <TryCount/>
    <is_managed/>
    <Enable_Event_Creation/>
    <Criticality/>
    <Alarm_On_Link_Down_Trap attribute_id="0x11fc2" />
    <Assert_Link_Down_Alarm attribute_id="0x12957" />
    <Utilization_Threshold attribute_id="0x1294b" />
    <Utilization_Reset attribute_id="0x1294f" />
    <Utilization_Threshold_Violation_Duration attribute_id="0x12be4" />
    <Inbound_Utilization_Threshold attribute_id="0x12d9f" />
    <Inbound_Utilization_Reset attribute_id="0x12da0" />
    <Inbound_Utilization_Threshold_Violation_Duration attribute_id="0x12da2"
    />
    <Outbound_Utilization_Threshold attribute_id="0x12da3" />
    <Outbound_Utilization_Reset attribute_id="0x12da4" />
    <Outbound_Utilization_Threshold_Violation_Duration attribute_id="0x12da6"
    />
    <Total_Packet_Rate_Threshold attribute_id="0x12da7" />
    <Total_Packet_Rate_Reset attribute_id="0x12da8" />
    <Total_Packet_Rate_Threshold_Violation_Duration attribute_id="0x12be3" />
    <Error_Rate_Threshold attribute_id="0x1294d" />
```

```
    <Error_Rate_Threshold_Reset attribute_id="0x12951" />
    <Error_Rate_Threshold_Violation_Duration attribute_id="0x12be5" />
    <Discarded_Threshold attribute_id="0x1294e" />
    <Discarded_Threshold_Reset attribute_id="0x12952" />
    <Discarded_Threshold_Violation_Duration attribute_id="0x12be2" />
</PortExportAttributes>

<SpectrumConfigurationExport model_type="VNM">
    <Minimum_Disk_Space attribute_id="0x119d2" />
    <Security_String/>
    <Unmanaged_Trap_Handling attribute_id="0x11cce" />
    <Trap_Storm_Rate attribute_id="0x122db" />
    <Trap_Storm_Length attribute_id="0x122da" />
    <Auto_Connects attribute_id="0x11f99"/>
    <Device_Thresholds attribute_id="0x12acd" />
    <Use_Full_Qualified_Host_Name attribute_id="0x12984" />
    <Allow_Non_Admin_SNMP_Community_Edit attribute_id="0x12042" />
    <Edit_Notes_By_Read_Only_User attribute_id="0x12043" />
    <Set_isManaged_By_Read_Only_User attribute_id="0x129f3" />
    <Consolidate_Users_In_Group attribute_id="0x12a1d" />
    <Copy_Users_When_Copying_Group attribute_id="0x12a5e" />
    <VLAN_Configuration attribute_id="0x129ad" />
    <Log_When_Device_Cannot_Be_Contacted attribute_id="0x12943" />
</SpectrumConfigurationExport>

<SpectrumConfigurationExport model_type="TopologyWrkSpc">
    <Create_WA_Link_Model attribute_id="0x25e0033" />
    <Create_LAN_IP_Subnet_Model attribute_id="0x25e000d" />
    <Create_Physical_Addresses attribute_id="0x25e000c" />
    <Create_Fanout_Models attribute_id="0x25e002e" />
    <Run_ATM_Discovery attribute_id="0x25e002d" />
    <IP_Route_Tables attribute_id="0x25e0006" />
    <Source_Addr_Tables attribute_id="0x25e0025" />
    <Spanning_Tree_Tables attribute_id="0x25e0026" />
    <Proprietary_Disc_Tables attribute_id="0x25e002b" />
    <ARP_Tables attribute_id="0x25e003a" />
    <Traffic_Resolution attribute_id="0x25e002f" />
    <Unmanaged_SNMP_Disc attribute_id="0x25e0034" />
    <New_Device_In_Maint_Mode attribute_id="0x25e0035" />
</SpectrumConfigurationExport>

<SpectrumConfigurationExport model_type="LostFound" >
    <Automatic_Model_Destruction attribute_id="0x11de1" />
    <Model_Destruction_Interval_Hours attribute_id="0x11de3" />
    <Model_Destruction_Interval_Minutes attribute_id="0x11de4" />
</SpectrumConfigurationExport>

<SpectrumConfigurationExport model_type="FaultIsolation">
    <ICMP_Support_Enabled attribute_id="0x11d98" />
    <ICMP_Timeout attribute_id="0x11dab" />
    <ICMP_TryCount attribute_id="0x11dac" />
```

```
    <Lost_Device_TryCount attribute_id="0x12a0a" />
    <Contact_Lost_Model_Destruction attribute_id="0x11fa8" />
    <Destruction_Delay attribute_id="0x11fa9" />
    <Destruction_Event_Generation attribute_id="0x11faa" />
    <Router_Redundancy_Retry_Count attribute_id="0x12a09" />
    <Port_Fault_Correlation attribute_id="0x129e6" />
    <Unresolved_Fault_Alarm_Disposition attribute_id="0x129f4" />
    <WA_Link_Fault_Isolation_Mode attribute_id="0x12adc" />
</SpectrumConfigurationExport>

<SpectrumConfigurationExport model_type="LivePipes" >
    <Live_Pipe_Enabled attribute_id="0x11df9" />
    <Alarm_Linked_Port attribute_id="0x11fbd" />
    <Suppress_Linked_Port_Alarms attribute_id="0x11fbe" />
    <Port_Always_Down_Alarm_Suppression attribute_id="0x129fb" />
</SpectrumConfigurationExport>

<SpectrumConfigurationExport model_type="AlarmMgmt" >
    <Generate_Alarm_Event attribute_id="0x11f5f" />
    <Add_Event_To_Alarms attribute_id="0x11f5c" />
    <Use_Old_Alarm_Event attribute_id="0x11f5d" />
    <Alarm_Update_by_Read_Only attribute_id="0x11f5e" />
    <Alarm_Ageout_Time attribute_id="0x129ea" />
    <Disable_Initial_Alarms attribute_id="0x11f5a" />
    <Disable_Suppressed_Alarms attribute_id="0x11f5b" />
    <Disable_Maint_Alarms attribute_id="0x11f59" />
    <Alarm_Clear_By_Read_Only attribute_id="0x11fb2" />
    <Ageout_Residual_Alarm_Only attribute_id="0x129ec" />
</SpectrumConfigurationExport>

<SpectrumConfigurationExport model_type="PolicyManager" >
    <Policy_Distribution_Mode attribute_id="0x4ad0007" />
</SpectrumConfigurationExport>

<SpectrumConfigurationExport model_type="GlobalConfig" >
    <SNMPv3Profiles attribute_id="0x12bd4" />
    <HibernationCommSuccessTries attribute_id="0x12acb" />
</SpectrumConfigurationExport>

</TopologyImportExportResourceFile>
```