

CA Spectrum®

模型化閘道工具組指南

9.4 版



本文件包含內嵌說明系統與文件 (以下稱為「文件」) 僅供您參考之用，且 CA 得隨時予以變更或撤銷。

未經 CA 事先書面同意，任何人不得對本「文件」之任何部份或全部內容進行影印、傳閱、再製、公開、修改或複製。此「文件」為 CA 之機密與專屬資訊，您不得予以洩漏或用於任何其他用途，除非 (i) 您與 CA 已另立協議管理與本「文件」相關之 CA 軟體之使用；或 (ii) 與 CA 另立保密協議同意使用之用途。

即便上述，若您為「文件」中所列軟體產品之授權使用者，則可列印或提供合理份數之「文件」複本，供您以及您的員工內部用於與該軟體相關之用途，但每份再製複本均須附上所有 CA 的版權聲明與說明。

列印或提供「文件」複本之權利僅限於軟體的相關授權有效期間。如果該授權因任何原因而終止，您有責任向 CA 以書面證明該「文件」的所有複本與部份複本均已經交還 CA 或銷毀。

在相關法律許可的情況下，CA 係依「現狀」提供本文件且不做任何形式之保證，其包括但不限於任何針對商品適銷性、適用於特定目的或不侵權的暗示保證。在任何情況下，CA 對於您或任何第三方由於使用本文件而引起的直接、間接損失或傷害，其包括但不限於利潤損失、投資損失、業務中斷、商譽損失或資料遺失，即使 CA 已被明確告知此類損失或損害的可能性，CA 均毋須負責。

「文件」中提及之任何軟體產品的使用均須遵守相關授權協議之規定，本聲明中任何條款均不得將其修改之。

此「文件」的製造商為 CA。

僅授與「有限權利」。美國政府對其之使用、複製或公開皆受 FAR 條款 12.212，52.227-14 與 52.227-19(c)(1) - (2) 與 DFARS 條款 252.227-7014(b)(3) 中所設之相關條款或其後續條約之限制。

Copyright © 2014 CA. All rights reserved. 本文提及的所有商標、商品名稱、服務標章和公司標誌均為相關公司所有。

CA Technologies 產品參考資料

本文件提及下列 CA Technologies 產品：

- CA Spectrum® (CA Spectrum)
- CA Spectrum® 模型化閘道工具組 (模型化閘道)
- CA CMDB

連絡技術支援

如需線上技術協助及完整的地址清單、主要服務時間以及電話號碼，請洽「技術支援」，網址為：<http://www.ca.com/worldwide>。

目錄

第 1 章：模型化閘道總覽	9
模型化閘道先決條件.....	9
關於模型化閘道工具組.....	9
匯入架構.....	10
匯出架構.....	12
第 2 章：將拓撲資料匯入到 CA Spectrum	13
如何使用模型化閘道進行匯入.....	13
將資料格式化成輸入檔.....	14
XML 輸入檔.....	14
XML 輸入檔語法.....	17
逗號分隔輸入檔.....	29
逗號分隔輸入檔語法.....	30
執行 modelinggateway 工具來匯入.....	30
ImportConfiguration 元素.....	31
匯入逗號分隔檔.....	32
檢視匯入資訊.....	32
在 OneClick 中檢視模型化閘道結果.....	33
錯誤記錄.....	34
第 3 章：從 CA Spectrum 匯出拓撲資料	35
關於從 CA Spectrum 匯出拓撲資料.....	35
配置匯出設定.....	36
ExportConfiguration 元素.....	37
用於匯出的 modelinggateway 工具.....	39
匯出 CA Spectrum 拓撲資料.....	40
匯入模型化閘道 XML 檔.....	40
附錄 A：文件類型定義元素	41
關聯.....	41
連線.....	42
相關.....	42
Correlation_Domain.....	43

CustomerManager	44
終結.....	44
裝置.....	45
EventModel	48
GenericView	50
GenericView_Container	51
GlobalCollection	52
匯入.....	53
Left_Model.....	54
List_Value	54
位置.....	55
Location_Container	55
Model_Attr.....	57
模型.....	58
MonitorPolicy_Attr.....	58
連接埠.....	59
Right_Model.....	61
RTM_Test	62
排程.....	62
SM_AttrMonitor	66
SM_Customer.....	67
SM_CustomerGroup.....	69
SM_Guarantee	70
SM_LatencyMon	71
SM_Service	72
SM_Service_Mgt	73
SM_ServiceMgr	74
SM_SLA.....	75
SM_SLA_Mgr	76
拓撲.....	77
Topology_Container	77
更新.....	79

附錄 B：文件類型定義檔 81

附錄 C：XML 範例 101

範例 1：匯入到拓撲檢視.....	101
範例 2：建立連線.....	102
範例 3：更新和終結.....	103
範例 4：建立、更新和終結.....	105

第 1 章：模型化閘道總覽

本節包含以下主題：

[模型化閘道先決條件](#) (位於 p. 9)

[關於模型化閘道工具組](#) (位於 p. 9)

[匯入架構](#) (位於 p. 10)

[匯出架構](#) (位於 p. 12)

模型化閘道先決條件

使用 CA Spectrum 模型化閘道工具組之前，請確定您符合下列條件：

- 大幅接觸過 CA Spectrum。
- 閱讀過《*概念指南*》。
- 有使用 XML 的知識。
- 瞭解文件類型定義 (DTD) 的概念。
- 對於您要匯入的網路拓撲有深入瞭解。
- 能夠使用 UNIX 或 Windows 來瀏覽檔案系統、建立和刪除檔案，以及建立和編輯文字檔。

關於模型化閘道工具組

CA Spectrum 模型化閘道工具組可讓整合者在 CA Spectrum 中匯入和匯出網路拓撲。此工具組包含用來定義 XML 元素和屬性的文字類型定義 (DTD)。此工具組也包含資源檔，用來定義 CA Spectrum 語法及要匯入或匯出的資訊。

使用 DTD 元素來匯入拓撲時，您可以建立 XML 檔來描述您網路上的裝置、連接埠和連線。此 XML 檔可以在 CA Spectrum 中建立新的拓撲、更新現有的資料，或終結不再正確的資料。此外，XML 語法中使用的元素和屬性也可以擴充和自訂，以符合大多數整合的需求。

此工具組也可讓您使用逗號分隔 ASCII 文字檔來匯入框架轉送或 ATM 連線。您也可以使用先前提及的 XML 功能來匯入這些連線資訊。

當 CA Spectrum 中有網路拓撲資料時，您就可以管理這些裝置，就像管理其他任何手動建立或以 Discovery 建立的模型一樣。您可以檢視匯入的結果，以及每次匯入的任何診斷資訊。

模型化閘道工具組也可讓您使用 XML 檔，從 CA Spectrum 匯出拓撲資訊和設定設定。然後，您就可以透過模型化閘道，將這些資訊匯入到指定的 SpectroSERVER。

在過去，要持續在 CA Spectrum 中填入動態網路拓撲資訊是件困難的工作。Discovery 和手動模型化並不適合用來進行變動環境中所需的持續更新。在各種不同的實體基礎結構中(例如下列環境)，使用 Discovery 將連線模型化也是一項挑戰，：

- 有線 MSO (Multi-Service Operator，有線電視服務業者)
- ATM (Asynchronous Transfer Mode，非同步傳輸模式)
- 框架轉送

對於這些問題，CA Spectrum 模型化閘道是有效的解決方案。

匯入架構

以匯入來說，在匯入整合過程中，您會從第三方資料庫取得資料，並建立輸入檔。根據內容而定，此輸入檔可以是 XML 檔或逗號分隔 ASCII 檔。XML 輸入檔提供最廣泛的匯入選項，也是本指南的討論重點。逗號分隔檔可讓您建立框架轉送和 ATM 電路的連線。

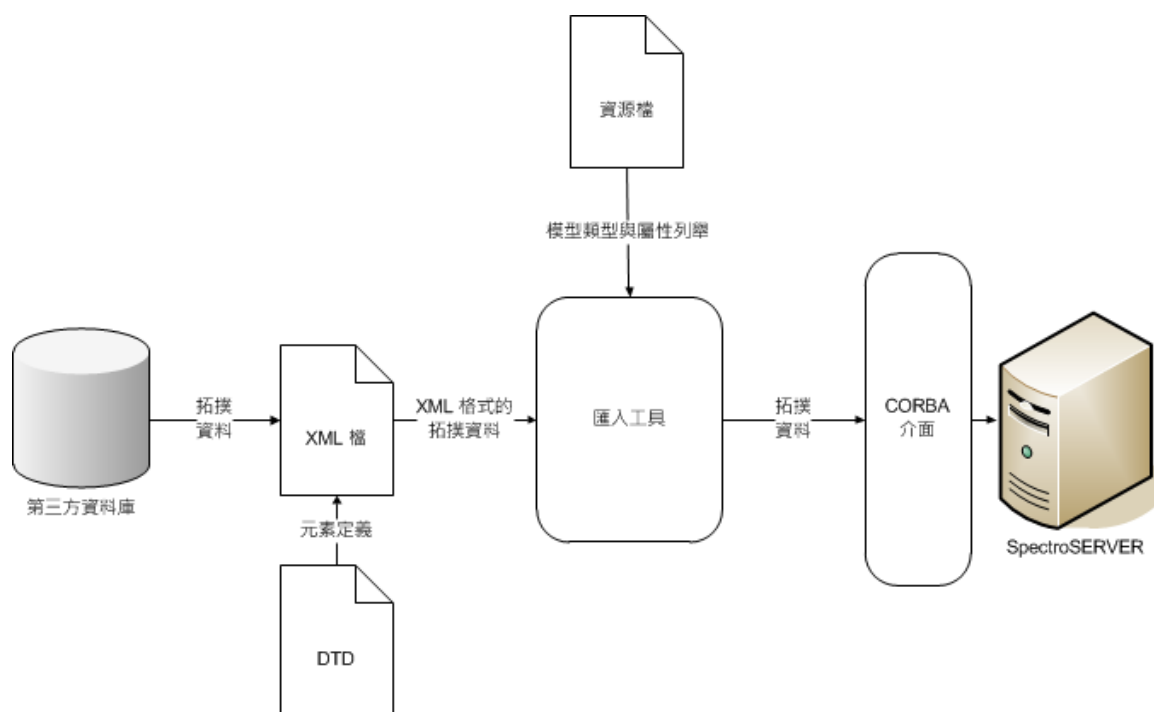
建立 XML 輸入檔時，請使用提供的文件類型定義 (DTD) 檔和 .modelinggatewayresource.xml 檔。DTD 定義 XML 元素、屬性及其相關語法規則。 .modelinggatewayresource.xml 檔顯示可用的 CA Spectrum 模型類型和屬性。此檔案會將 CA Spectrum 模型類型名稱和屬性名稱，與 CA Spectrum 用於該模型類型或屬性的獨特十六進位識別碼相連。您可以自訂 .modelinggatewayresource.xml 檔，以符合特定的整合需求。

建立第一個輸入檔之後，即可將之當成範本來建立多個代表相同輸入類型的資料集。例如，您可以建立某個 XML 檔來匯入裝置，而以後只要替換成裝置特定的拓撲資料，就可重複使用此檔案。

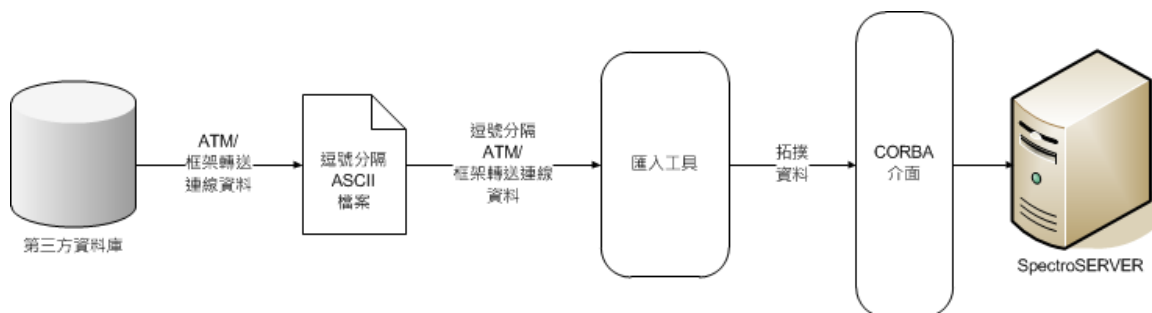
`modelinggateway` 工具是命令列公用程式，其會讀取輸入檔中的網路拓撲資訊，並將資料傳送至 SpectroSERVER 資料庫。有了 XML 檔中的資料後，匯入工具就可以建立、終結及更新連線、裝置和容器模型。此工具也可以從 CA Spectrum 匯出資料。

CA Spectrum 模型化閘道也有機制可驗證每次匯入資料庫時的安全性和正確性。例如，其可保存稽核追蹤，包括每次進行建立、刪除、關聯和更新所留下的記錄。您可以在 OneClick 內檢視匯入作業的相關資訊。當匯入過程中發生任何類型的重大失敗時，CA Spectrum 會產生事件來報告錯誤。所有錯誤及可能的原因都記錄在錯誤記錄檔中。您也可以開啓偵錯記錄，協助您查明問題來源或不正確的地方。

下圖顯示模型化閘道如何使用 XML 檔來匯入資料。資料從第三方資料庫流出，並根據 DTD 和 `.modelinggatewayresource.xml` 編成正確語法格式的 XML 檔。然後，匯入工具解譯 XML 檔，並透過 CORBA 介面將資料傳送至 CA Spectrum。



下圖顯示模型化閘道如何使用逗號分隔 ASCII 文字檔來匯入框架轉送和 ATM 連線資訊。



匯出架構

以匯出 (使用 SpectroSERVER 做為資源) 而言，模型化閘道可以將拓撲和設定資料匯出到 XML 檔。然後，此 XML 檔就可以與第三方工具整合，或重新匯入到另一個 SpectroSERVER。例如，如果 CA Spectrum 磁碟分割或範圍控制代碼變更，可以重新匯入此檔案。

第 2 章：將拓撲資料匯入到 CA Spectrum

本節包含以下主題：

- [如何使用模型化閘道進行匯入](#) (位於 p. 13)
- [將資料格式化成輸入檔](#) (位於 p. 14)
- [執行 modelinggateway 工具來匯入](#) (位於 p. 30)
- [ImportConfiguration 元素](#) (位於 p. 31)
- [匯入逗號分隔檔](#) (位於 p. 32)
- [檢視匯入資訊](#) (位於 p. 32)

如何使用模型化閘道進行匯入

您可以使用模型化閘道將第三方拓撲資料匯入到 CA Spectrum。首先，從第三方資料庫擷取拓撲資料，再將資料格式化成輸入檔。

若要使用模型化閘道將第三方拓撲資料匯入到 CA Spectrum，請執行下列工作：

1. 擷取拓撲資料。

從第三方資料庫擷取網路拓撲資料。由於每個資料庫系統都不同，請參閱您所使用的資料庫的文件，以完成此步驟。

2. [將資料格式化成輸入檔](#) (位於 p. 14)。

若要為匯入工具來格式化資料，請建立 XML 或逗號分隔輸入檔。

3. (選用) 移動 modelinggateway 工具。

附註：若要在另一部伺服器上執行 modelinggateway 工具，請將 modelinggateway 工具及其所有支援檔案移至該伺服器。如需詳細資訊，請參閱《*分散式 SpectroSERVER 管理指南*》。

4. [執行 modelinggateway 工具](#) (位於 p. 30)。

建立輸入檔之後，使用匯入工具將資料傳送至 CA Spectrum。

5. (選用) [匯入逗號分隔檔](#) (位於 p. 32)。

您可以從 OneClick 介面匯入框架轉送和 ATM 連線，而不需要使用匯入工具。

6. [檢視匯入資訊](#) (位於 p. 32)

在 OneClick 中確認匯入的進度和結果。

附註：請勿使用模型化閘道將某個版本的 CA Spectrum 中的模型移轉到另一個版本。不同版本的 CA Spectrum 可能使用不同的方法來識別和模型化實體。因此，請勿使用模型化閘道來匯入任何從不同版本的 CA Spectrum 匯出的 XML 檔。

將資料格式化成輸入檔

模型化閘道使用兩種輸入檔來匯入資料：**XML 檔**和**逗號分隔檔**。**XML 輸入檔**可用來建立或終結模型和連線，以及更新屬性值和連線資訊。模型化閘道提供之 **DTD** 中的語法，定義 **XML 輸入檔**中要使用的元素。**逗號分隔檔**只能用來建立 **ATM** 和**框架轉送連線**。下列各節說明如何建立每一種輸入檔。

XML 輸入檔

若要瞭解用來建立 **XML 輸入檔**的元素，請熟悉 **CA Spectrum** 用來模型化網路基礎架構的程序。下節概述此程序，並討論其如何套用至 **XML 輸入檔**中使用的 **XML 元素**。如果您已熟悉這些概念，可跳過本節，繼續閱讀「[XML 輸入檔語法](#) (位於 p. 17)」。

更多資訊：

[XML 輸入檔語法](#) (位於 p. 17)

階層式檢視

CA Spectrum 中的檢視可用來組織資料，以利顯示或操作資料。階層式檢視可代表網路資料的結構方式。在 **XML 檔**中建立網路資料的結構時，您會選擇元素來代表每一個階層式檢視。有兩種階層式檢視：「**拓撲**」和「**位置**」。

拓撲檢視

「拓撲」檢視其實是網路元件的抽象概念。使用此檢視時，您會根據心裡歸納出的邏輯連線，來呈現網路的實體或邏輯元件，並將這些元件進行分組。您也可以選擇透過圖形來呈現連線，也就是在連接埠或裝置層級使用管道來顯示裝置的連接方式。在 OneClick 中，此檢視顯示為通用拓撲。

附註：如需 OneClick 中可用的拓撲檢視的詳細資訊，請參閱《*模型化與管理 IT 基礎結構管理員指南*》。

位置檢視

「位置」檢視依實體位置來組織您的網路資料。在此檢視中，您可以用地理位置來呈現您的網路。您可以從全球辦事處開始。然後，直接進入您的辦事處所在的每個地區中每棟建築物每個樓層的佈線室。在 OneClick 中，此檢視顯示為 World 拓撲。

附註：如需 OneClick 中可用的拓撲檢視的詳細資訊，請參閱《*模型化與管理 IT 基礎結構管理員指南*》。

模型和模型類型

CA Spectrum 中預先定義許多模型類型。當模型類型在 CA Spectrum 介面中具現化來代表特定網路實體時，就稱為模型。模型類型的兩個主要類別為：

- 智慧模型類型
- 容器模型類型

智慧模型類型可具現化來代表網路上運作的實際裝置。它們有 IP 和 MAC 位址，而 CA Spectrum 可以使用 SNMP 直接與這些裝置通訊。容器模型類型會具現化成的模型，主要用來將模型分組在一起。

您可以根據使用的階層式檢視類型，將模型分組。例如，您可以使用拓撲檢視來建立 LAN 模型，來將您網路某個區段上的特定裝置分組。或者，您可以使用位置檢視來建立 Room 模型，以將建築物某間機房中的裝置分組。

容器模型可以包含其他容器模型和/或智慧模型，視特定模型類型而定。例如，網路容器模型可能包含智慧模型來代表路由器。網路容器模型也可以包含 LAN 容器模型來代表 IP 位址範圍。另一方面，Building 模型只能包含容器模型，例如 Floor、Section 或 Room。

DTD 中的元素可讓您使用任何階層式檢視及其相關容器模型類型，來呈現您的網路拓撲。您也可以使用任何可具現化的智慧模型類型。智慧模型類型並不依賴所使用的階層式檢視類型。

附註：如有需要，您可以指定模型控制代碼而非模型類型來識別模型。指定模型控制代碼時，模型化閘道會忽略其他任何模型識別碼，只使用模型控制代碼來識別模型。

CA Spectrum 知識庫中定義的模型類型，並非全部都可以實際用來在 OneClick 中建立模型。有些是當成用來衍生其他模型類型的基礎模型類型。

附註：如需詳細資訊，請參閱《概念指南》。

模型化的方法

CA Spectrum 中使用兩種方法來模型化裝置。第一種方法是使用裝置的 IP 位址或 DNS 名稱。CA Spectrum 會利用這些資訊來聯繫裝置，並使用最能代表該裝置之功能的模型類型來建立模型。

第二種方法是提供模型類型來建立裝置模型。您仍然必須提供 IP 位址或 DNS 名稱，讓 CA Spectrum 能夠與裝置通訊。然而，無論 CA Spectrum 評估該裝置的功能為何，都會具現化您所選擇的模型類型或模型控制代碼。

若要建立非裝置的模型，例如容器模型，則必須提供模型類型和模型名稱進行匯入。

CA Spectrum 屬性

每個模型類型都有一組相關聯的屬性。每個屬性皆描述模型類型的某個方面。具現化的模型中的屬性具有值，可反映模型所代表的裝置，並說明模型的目前狀態。例如，模型類型 Host_Sun 具有屬性 IPAddress。如果 Host_Sun 類型的模型具現化，則此屬性的值會反映模型所代表之裝置的 IP 位址。

XML 語法也使用屬性一詞。XML 屬性描述元素的詳細資訊。在 CA Spectrum 模型化閘道 XML 語法中，有些 XML 屬性用來提供 CA Spectrum 屬性的值。

附註：CA Spectrum 所定義的屬性稱為 CA Spectrum 屬性；一般 XML 屬性則簡單稱為屬性。

XML 輸入檔語法

產生 XML 檔時，請使用 DTD 檔中定義的語法規則。下節概述每個 DTD 元素的功能。

下列說明和範例並未涵蓋每個元素的所有屬性。如需每個元素及其屬性的完整參考資料，請參閱「[文件類型定義元素](#) (位於 p. 41)」。

根元素

DTD 中定義的元素以階層式結構存在，此階層式結構與 CA Spectrum 內的網路表示法相呼應。每個 XML 匯入檔中必須使用的根元素是 **Import** 元素。XML 語法規則指定根元素是最外圍的元素，表示 XML 檔的開頭和結尾。因此，**Import** 元素括住您文件中使用的其餘 XML 元素。

模型導向元素

模型導向元素定義您網路的實體或邏輯元件。它們是容器類型元素，用來建立模型，以定義網路元素的邏輯分組方式。分組時是根據元素所在 CA Spectrum 階層式檢視的類型進行。這些容器類型元素每個皆可以存在於其中一個特定的階層式檢視中。

- Topology_Container
- Location_Container
- 裝置
- 排程
- 連接埠
- 連線
- GenericView_Container

Topology_Container

Topology_Container 元素建立的模型會根據實體或邏輯拓撲將其他模型分組。Topology_Container 元素會建立容器模型，因此請使用 **model_type** 屬性或模型控制代碼，以識別您要使用的特定容器。DTD 中會列舉可能的 **model_type** 值。LAN 就是 Topology_Container **model_type** 值的例子。您使用 **name** 屬性來指定 Topology_Container 的名稱。**name** 和 **model_type** 屬性可唯一識別所建立的模型。如果您是指定模型控制代碼，則系統會忽略 **name** 和 **model_type** 屬性。Topology_Container 可以包含其他 Topology_Container 元素、裝置或連線。Topology_Container 模型一律放在 OneClick 拓撲檢視中。

附註：依預設，`name` 和 `model_type` 屬性會提供值給 `CA Spectrum` 屬性 `Model_Name` 和 `Modeltype_Name`。不過，您可以編輯 `.modelinggatewayresource.xml` 檔，以變更由 `name` 屬性來提供值的 `CA Spectrum` 屬性。無論選擇什麼新的 `CA Spectrum` 屬性 (以及模型類型)，都會用來唯一識別容器。此變更讓兩個容器有相同的模型名稱。

Location_Container

`Location_Container` 會根據實體或地理位置將其他模型分組。`Building` 和 `Room` 都是 `Location_Container` 元素模型類型值的例子。

`Location_Container` 元素會建立容器模型，因此請使用 `model_type` 屬性或模型控制代碼，以識別您要使用的特定容器。`DTD` 中會列舉可能的 `model_type` 值。您是使用 `name` 屬性來指定 `Location_Container` 的名稱。`name` 和 `model_type` 屬性可唯一識別所建立的模型。如果您是指定模型控制代碼，則系統會忽略 `name` 和 `model_type` 屬性。

附註：依預設，`name` 和 `model_type` 屬性會提供值給 `CA Spectrum` 屬性 `Model_Name` 和 `Modeltype_Name`。不過，您可以編輯 `.modelinggatewayresource.xml` 檔，以變更由 `name` 屬性來提供值的 `CA Spectrum` 屬性。無論選擇什麼新的 `CA Spectrum` 屬性 (以及模型類型)，都會用來唯一識別容器。此變更讓兩個容器有相同的模型名稱。

裝置

`Device` 元素定義網路上的某個裝置。此元素與其他元素一起使用，在 `CA Spectrum` 中建立、更新或終結裝置模型的某個執行個體。處理 `SNMP` 裝置時，請在 `ip_dnsname` 屬性中提供有效和唯一的 IP 位址或 DNS 名稱來唯一識別裝置。此唯一識別可讓 `CA Spectrum` 與裝置通訊，並根據裝置的功能選取最適合的模型類型。請將 `ip_dnsname` 設定為有效字串。如果 `ip_dnsname` 無效或無法聯繫，則可能會無法建立裝置模型。如果提供的 `model_type` 是搭配無效或無法聯繫的 `ip_dnsname`，則仍然會使用指定的模型類型來建立裝置模型。但是，系統不會啟動該裝置模型來提供任何有效的網路資訊或狀態。`.modelinggatewayresource.xml` 中會列舉裝置可能的 `model_type` 值。

排程

`Schedule` 元素定義何時讓裝置模型進入維護模式。當裝置處於維護模式時，對裝置及其元件的管理流量會暫停。暫停流量會使得 `CA Spectrum` 在您對裝置進行維護的期間，無法在裝置模型上產生任何事件或警報。

連接埠

建立裝置模型時會自動建立裝置的連接埠。Port 元素可讓您修改一些 CA Spectrum 連接埠屬性值，或指定連接埠層級的連線。您可以指定不同類型的連接埠，例如框架轉送或 ATM 電路。若要識別裝置上的連接埠，請提供 identifier_name 和 identifier_value 屬性的值。DTD 中會列舉 identifier_name 可能的值。identifier_value 是 identifier_name 屬性所選擇之識別碼的值。Port 元素必須一律當成 Device 的下層元素來指定。

連線

Connection 元素定義兩個裝置之間的實體或邏輯連線 (包括 WAN 連結連線)，因此必須包含兩個 Device 下層元素。如果 Device 元素中指定 Port 元素，則會在該裝置的指定連接埠上解析連線。如果 Device 元素未指定 Port 元素，CA Spectrum Discovery 會嘗試在要解析的連線中決定連接埠。

GenericView_Container

若要在一般檢視中建立容器模型，請使用 GenericView_Container 元素。GenericView 和 GenericView_Container 元素都可用來建立自訂的檢視。因此，身為整合者的您，應該決定何時或如何使用此容器。

工作導向元素

DTD 中定義的其餘元素都是工作導向元素。這些元素及其屬性可協助定義輸入檔所產生的動作類型。利用這些元素，您可以建立新的拓撲資訊，以及更新、覆寫或刪除現有的拓撲資訊。個別輸入檔可以使用這其中零個或一個元素，但 Connection 元素除外。您可以視需要使用任何數量的 Connection 元素。

- 拓撲
- 位置
- GenericView
- 連線
- 更新
- 終結

新增拓撲資料

工作導向元素會定義您要對 XML 檔採取的動作。想要在 CA Spectrum 中建立新的網路拓撲資料時，請使用 **Topology** 和 **Location** 元素。這些元素定義您想在其中建立資料的階層式檢視。然後，您可以使用對應的模型元素做為下層元素，以建立網路實體的模型。若要根據特定的整合需求來自訂檢視，請使用 **GenericView** 元素。

建立位置檢視

您可以在位置檢視中建立拓撲資訊，以便於 OneClick 的 World 拓撲中檢視。若要在位置檢視中建立此資訊，請在 **Import** 根元素內使用 **Location** 元素來建構 XML 檔。Location 元素可以包含 **Location_Container** 元素以建立特定的容器模型，或包含 **Device** 元素以建立裝置模型。

範例：位置檢視中的 Site 容器

下列範例會在位置檢視中建立 Site 容器，並在該容器內建立裝置。

```
<?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 元素是輸入檔中一定要包含的根元素。

Location 元素表示您在位置檢視中建立模型。

Location_Container 元素建立容器模型。此模型是邏輯元件，而非網路的實體元件。因此，CA Spectrum 無法聯繫它，也無法使用 IP 位址或 DNS 名稱來定義模型類型。若要指出要建立的容器模型類型，請提供 **model_type** 屬性和 **name** 屬性的值。DTD 和後續的 [Location Container](#) (位於 p. 55) 一節中會列出可能的 **model_type** 屬性值。**name** 是必要屬性，且必須指定模型的唯一名稱。

附註：指定模型控制代碼時，則會使用模型控制代碼來識別容器模型。因此，如有提供 **name** 和 **model_type** 屬性，則會加以忽略。

Device 元素在 Site Location_Container 模型內建立模型。ip_dnsname 屬性是 Device 元素的必要屬性。如果可聯繫裝置，CA Spectrum 會使用 IP 位置或 DNS 名稱來尋找該裝置。

如需這些元素及其可能屬性的完整參考資料，請參閱「[文件類型定義元素](#) (位於 p. 41)」。

建立拓撲檢視

您可以將網路資料匯入到拓撲檢視，以便於 OneClick 的通用拓撲中檢視。若要匯入到拓撲檢視，請在 Import 根元素內使用 Topology 元素來建構 XML 檔。Topology 元素可以包含：

- Topology_Container 元素，以建立特定類型的容器模型。
- Device 元素，以建立特定類型的裝置模型。
- Connection 元素，以建立兩個裝置之間的連線。

您可以使用 DTD 中的階層和語法規則，正確表達網路的實體和邏輯連線。

範例：拓撲檢視中的 LAN 容器

此範例會在拓撲檢視中建立 LAN 容器，並在該容器內建立裝置。

```
<?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 元素是輸入檔中一定要包含的根元素。

Topology 元素表示您要在拓撲檢視中建立模型。

Topology_Container 元素建立容器模型。此模型是邏輯元件，而非網路的實體元件。因此，CA Spectrum 無法聯繫它，也無法使用 IP 位址或 DNS 名稱來定義模型類型。若要指出要建立的容器模型類型，請提供 **model_type** 屬性和 **name** 屬性的值。DTD 和後續的 [Topology Container](#) (位於 p. 77) 一節中會列出可能的 **model_type** 屬性值。**name** 是必要屬性，且必須指定模型的唯一名稱。指定的其他屬性是選用屬性。

附註：指定模型控制代碼時，則會使用模型控制代碼來識別容器模型。因此，如有提供 **name** 和 **model_type** 屬性，則會加以忽略。

Device 元素在 LAN **Topology_Container** 模型內建立模型。**ip_dnsname** 屬性是 **Device** 元素的必要屬性。CA Spectrum 如果可以聯繫裝置，則會使用 IP 位置或 DNS 名稱來尋找該裝置。CA Spectrum 在找到裝置時，會決定用來建立模型的適當模型類型。

在多個檢視中表示相同裝置

您可以建立 XML 檔，並讓其在多個檢視中代表一個裝置。若要建立此檔案，則建議您每個用來為此裝置建立模型的 **Device** 元素，都具有相同的屬性和屬性值。如果不相同，匯入工具會嘗試合併這些 **Device** 元素的屬性和值，以建立一組一致的屬性和值。有時，每個 **Device** 元素中都指定了某個屬性，但使用不同的值。在此情況下，XML 檔中列出的最後一個 **Device** 元素的值，在用來建立該裝置的模型的其他 **Device** 元素中，將覆寫該屬性的所有先前的值。

請記住，有些屬性具有預設值。例如，**community_string** 屬性的預設值是 **public**。因此，指定 **Device** 屬性和值來代表裝置 A 時，建議也在其他任何代表裝置 A 的 **Device** 元素中指定此值。如此可確保該屬性的預設值不會用來覆寫先前指定的值。

範例：同時在拓撲和位置檢視中建立裝置

下列範例會在拓撲和位置檢視中建立裝置 10.253.9.16。您可以看到這兩個檢視中用來描述裝置的屬性和值都相同。

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- Topology View import -->
    <Topology discover_connections="false" complete_topology="false">
        <Device ip_dnsname="10.253.9.16" community_string="zippo" />
    </Topology>
</Import>
```

```

<!-- Location View import -->
  <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 Discovery 建立連線

CA Spectrum 連線代表兩個裝置之間的實體或邏輯連結。在 XML 輸入檔中，有兩種不同方法可以建立連線。第一種方法運用 CA Spectrum Discovery，使用 Topology 元素的 discover_connections 屬性。當 discover_connections 屬性設定為 true 時，Discovery 會在新建立的裝置模型上執行。然後，Discovery 會建立這些裝置的連線。

附註：如需 Discovery 的詳細資訊，請參閱《*模型化與管理 IT 基礎結構管理員指南*》。

範例：使用 Discovery 建立連線

```

<?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 元素建立連線

第二種建立連線的方法是使用 Connection 元素，以連接已建立的裝置。您可以指定兩個連接埠之間、一個裝置與一個連接埠之間，或兩個裝置之間的連線。

指定裝置之間的連線可讓 CA Spectrum 隔離裝置的錯誤，但最好指定連接埠層級的連線。連接埠層級連線是更細等級的連線，可讓 CA Spectrum 從連接埠層級來解析連線和分析錯誤。當您指定兩個裝置之間的連線，但未指定連線中使用的一端或兩端連接埠時，CA Spectrum 會自動嘗試決定連接埠。如果此過程順利，CA Spectrum 會解析連線到連接埠層級。

同時發生下列兩種狀況時，CA Spectrum 會產生錯誤來指出連線錯誤：

- CA Spectrum 無法決定連線中使用的兩端連接埠。
- 這些裝置中至少有一個是可管理的裝置。

錯誤會寫入錯誤記錄檔中。

如果 CA Spectrum 只能決定其中一端連接埠，則只會在連線的一端解析連線到連接埠層級。另一端則維持解析到裝置層級。

如果兩個裝置都是無法管理的裝置，CA Spectrum 會建立裝置層級的連線。

範例：建立現有連接埠之間的連線

下列範例會建立兩個現有連接埠之間的連線，每個連接埠屬於不同的裝置。Connection 元素同時識別要連結的 Port 和 Device 元素。因為每個 Device 元素內都指定 Port 元素，所以連線在這兩個裝置都是解析到連接埠層級。

```
<?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>
```

上述範例指定 DLCI 連接埠之間的連線。因為 identifier_name 屬性的值是 frCircuitTableInstance，所以連接埠是以 MIB 中 frCircuitTable 物件內的 OID 執行個體值來識別。identifier_value 屬性用來指定 OID 執行個體值。

Connection 元素位在 Topology 元素或 Topology_Container 元素內，以指出裝置的階層位置。此情況不會變更輸入檔的結果。

重要！ 當您嘗試匯入的 XML 檔中有多個 **Connection** 元素使用相同的 **Port** 元素時，模型化閘道不會報告錯誤。一個連接埠不能有多個連線。如果多個 **Connection** 元素中指定相同的連接埠，則 XML 檔中最後一個 **Connection** 元素會覆寫所有先前指定該連接埠的 **Connection** 元素。

更多資訊：

[錯誤記錄](#) (位於 p. 34)

建立 WA_Link 連線

若要建立 WA_Link 連線，請使用下列語法：

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

模型化閘道會自動建立 **WA_Segment**。區段和一個或多個裝置之間會建立連結。若要指定第二個裝置和連結之間的連線，請將第二個連線標籤新增至匯入檔。

同步化 CA Spectrum 與第三方資料庫之間的資訊

Topology、**Location**、**Topology_Container** 及 **Location_Container** 元素都有名為 **complete_topology** 的屬性。將此屬性的值設定為 **true**，表示 XML 檔定義 **CA Spectrum** 必須瞭解的所有模型和連線。將 XML 檔匯入到 **CA Spectrum** 時，會將任何存在於該 **CA Spectrum** 檢視中，但在 XML 檔中沒有相對應表示的模型傳送到「尋獲失物」。如果檢視中有子容器，**CA Spectrum** 會參考指定子容器的元素中所設定的 **complete_topology** 屬性的值。如果子容器元素中未指定 **complete_topology** 屬性值，則值繼承自上層元素。因此，如果上層元素的 **complete_topology** 設定為 **true**，而子容器元素未指定 **complete_topology** 的設定，則子容器的 **complete_topology** 值也是 **true**。

當 **CA Spectrum** 匯入 XML 檔時，在這些情況下會將模型傳送到「尋獲失物」：

- 模型直接存在於您匯入至的檢視中，或存在於該檢視的子容器中。
- 模型不存在於 XML 檔中。

同步化第三方資料庫中的資料與 **CA Spectrum** 中的資料時，這種行為很有用。

範例：Complete_Topology 設定為 True

在下列範例中，Topology 元素內的 complete_topology 設定為 true。此輸入檔中指定的模型除外，所有存在於拓撲檢視中的模型都會傳送到「尋獲失物」。在此範例輸入檔中，只指定了兩個模型：

- LAN Topology_Container
- IP 位址為 10.253.9.18 的裝置

如果這些模型不存在，則會建立。如果存在，CA Spectrum 會使用輸入檔中的屬性值來更新其 CA Spectrum 屬性值。拓撲檢視中存在的其他任何模型 (VNM 除外) 都會傳送到「尋獲失物」。

```
<?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 complete_topology 屬性用在 Topology_Container 元素中，而不是用在 Topology 元素中，則 CA Spectrum 只會從該 Topology_Container 沿著階層往下移除未指定的模型。

更新資訊

若要更新現有模型的 CA Spectrum 屬性和關聯資訊，請使用 Update 元素。Update 元素可以括住 Container 元素、Device 元素及 Association 元素。Port 屬性的值可使用適當的 Device 元素來更新。

範例：更新兩個不同模型的屬性和建立關聯

下列範例顯示 **Update** 輸入檔。此案例中會更新兩個不同模型的兩個屬性，並建立兩個模型之間的關聯。

```
<?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>
```

第一個更新的屬性是 LAN 容器模型的 `model_name` 屬性。模型名稱從 `Sample` 變更為 `newLAN`。請注意其中使用了下列屬性：`name` 和 `model_name`。這兩個屬性的存在都是為了變更 CA Spectrum 屬性 `Model_Name`。若要識別容器模型，然後使用 `model_name` 屬性來指定容器模型的新名稱，請使用 `name` 屬性且指定目前名稱做為值。

接著，範例將裝置的輪詢間隔值從 `Test1` 變更為 `1108`。指派新的值給 `poll_interval` 屬性會覆寫舊的值。

只要 `Network` 模型類型的容器模型 `"Net"` 與裝置模型 `172.24.94.94` 都存在於 `SpectroSERVER` 中，此範例也會在這兩者之間建立關係 `0x10002` 的關聯。

終結資訊

使用 `Destroy` 元素來刪除容器模型、裝置模型、連線及關聯。終結裝置時，也會終結與該裝置相關聯的所有連接埠和應用程式模型。

範例：終結 LAN 容器、連線及關聯

下列範例會終結名為 `newLAN` 的 LAN 拓撲容器。此容器內的所有模型 (除非被指定要終結) 都會傳送到「尋獲失物」。此範例也會終結 `Test1` 與 `Test2` 兩個裝置之間以連接埠層級指定的連線。如果模型類型 `Network` 的容器模型 `"Net"` 與裝置 `172.24.94.94` 之間存在 `Collects` 關聯，則會終結此關聯。

```
<?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！ 如果要終結的模型代表的是已從網路中移除的裝置，則在 XML 檔的 Destroy 元素中指定 Device 的 ip_dnsname 屬性時，請使用裝置的 DNS 名稱，而不是 IP 位址。從網路中移除裝置後，該裝置的 DNS 項目就不再存在。因此，模型化閘道將無法識別適當的模型來刪除。

.modelinggatewayresource.xml 檔

Topology_Container、Location_Container 和 Device 元素有 model_type 屬性，此屬性的值必須等於有效的 CA Spectrum 模型類型。CA Spectrum 使用十六進位數字來唯一識別模型類型。資源檔 .modelinggatewayresource.xml 中已列舉這些十六進位值。此檔案會將模型類型的文字值與唯一的十六進位識別碼配對。然後文字值便會顯示在 DTD 中。

DTD 中定義的許多屬性對應至 CA Spectrum 屬性。CA Spectrum 中使用十六進位數字來唯一識別 CA Spectrum 屬性。.modelinggatewayresource.xml 檔不使用 DTD 或 XML 檔中的這些十六進位值。此檔案是將 CA Spectrum 屬性的十六進位識別碼與更直覺的文字名稱設定。

.modelinggatewayresource.xml 檔的 ModelType 元素和 Attribute 元素都可自訂。

附註： modelinggatewayresource.xml 檔也用於從 CA Spectrum 匯出拓撲資料。如需詳細資訊，請參閱「[從 CA Spectrum 匯出拓撲資料](#) (位於 p. 35)」。

定義字元集編碼

CA Spectrum XML 輸入檔預設以 UTF-8 編碼。若要匯入特殊字元或外國語言，請在 XML 檔標頭中指定適當的字元集編碼，如下列範例所示。

範例：將輸入檔字元編碼設定為希臘文

下列範例顯示如何修改 XML 檔，將字元編碼設定為希臘文：

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

檢視 CA Spectrum 字元集編碼資訊

如果需要判斷 CA Spectrum 正在使用的字元集編碼，您可以從 OneClick 管理頁面中判斷。

檢視 CA Spectrum 字元集編碼資訊

1. 在 OneClick 首頁中按一下 [管理]。
[管理] 頁面隨即開啓。
2. 按一下左邊面板中的 [字元集]。
這時會開啓 [字元集編碼] 頁面，其中顯示編碼清單及其適用的語言。

逗號分隔輸入檔

模型化閘道可以在 XML 輸入檔中指定 ATM 和框架轉送連線。此工具組也可以接受逗號分隔 ASCII 文字檔中的 ATM 和框架轉送連線資訊。此檔案可用來匯入下列項目之間的連線資訊：

- 兩個 ATM 電路
- 兩個框架轉送電路
- ATM 與框架轉送電路

您可以指定在 OneClick 中建立即時管道來代表連線。相同的輸入檔中可以指定多個連線。

附註： 這些連線中的相關裝置模型必須事先存在於 CA Spectrum 中。

逗號分隔輸入檔語法

下列範例顯示逗號分隔輸入檔中使用的格式：

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

Device_IP

指定連線中的每個相關裝置的 IP 位址。必要。

OID

指定 frCircuitTable、atmVclTable 或 atmVplTable 的 OID 執行個體，以指定裝置上的電路連結。

CircuitName

(選用) 指定相關電路的名稱。

CircuitID

(選用) 指定相關電路的 ID。

Pipe

(選用) 有兩個可能的值：CREATE_PIPE 或 NO_CREATE_PIPE。如果值設定為 CREATE_PIPE，則會在指定的連線之間建立即時管道。如果值設定為 NO_CREATE_PIPE，則不會在指定的連線之間建立即時管道。如果未指定此參數的值，則會採用 CREATE_PIPE 的預設值。

範例：框架轉送電路之間指定的連線

下列範例顯示輸入檔，其中指定兩個框架轉送電路之間的連線。這兩個連接埠之間將建立即時管道。

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

執行 modelinggateway 工具來匯入

若要執行 modelinggateway 工具來匯入，請使用下列語法。

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 主機的名稱。

-i import_file

指定 XML 檔的名稱，此檔案包含以 `.modelinggateway.dtd` 進行編譯時的必要輸入資訊。

-o outputfile

(選用) 將錯誤資訊記錄到 `outputfile` 參數中指名的檔案。如果不使用此選項，則錯誤資訊會記錄到名為 `import_file.log` 的檔案。`Import_file` 是 XML 檔的名稱。

-debug debugfile

(選用) 指出您想在匯入過程中建立偵錯輸出檔。使用 `-debug` 選項時，您可以提供自己的偵錯檔名進行輸出。如果未提供 `debugfile` 的值，則偵錯檔名預設為 `import_file` 名稱，結尾是 `".debug"`。

附註：在執行模型化閘道的機器上，`-debug` 選項需要磁碟空間。例如，當 `import_file` 中的模型數量很大，或裝置模型的介面密度很高時，產生的偵錯輸出檔會很大。

ImportConfiguration 元素

若要在某些方面控制模型化閘道匯入資料的方式，請使用 `ImportConfiguration` 元素。

`ImportConfiguration` 元素有下列語法：

```
<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 是否處理容器元素下發現之先前已存在 CA Spectrum 中的裝置。

預設值：false

import_to_primary_ss_only

指定當主要 SpectroSERVER 停機時，模型化閘道是否應連線至次要 SpectroSERVER。

預設值：false

max_device_creation_threads

指定可以同時建立和啟動多少個裝置模型。

附註：將此值設定為高於 50 會導致太多 SNMP 流量。

預設值：50

匯入逗號分隔檔

上一節說明如何使用 `modelinggateway` 匯入工具來匯入輸入檔。您也可以從 OneClick 介面匯入框架轉送和 ATM 連線資料。

請依循下列步驟：

1. 在 [OneClick 主控台] 按一下適用的 VNM 模型。
2. 在 [元件詳細資料] 面板中，按一下 [資訊] 索引標籤。
3. 按一下 [邏輯連線匯入] 以展開區段。
4. 按一下 [匯入]，找出含您要匯入到 CA Spectrum 的資料的逗號分隔檔，然後按一下 [開啓]。

OneClick 開始匯入資料。出現 [匯入結果] 對話方塊，其中提供匯入成功的相關資訊。

檢視匯入資訊

CA Spectrum 模型化閘道有機制可協助確保每次匯入資料庫時的安全性和正確性。CA Spectrum 模型化閘道會保存稽核追蹤，包括每次進行建立、刪除、關聯和更新所留下的記錄。您可以在 OneClick 內檢視匯入作業的相關資料，也可以在錯誤和偵錯記錄中追蹤匯入問題的資訊。

在 OneClick 中檢視模型化閘道結果

您可以從 OneClick 主控台的 VNM 模型、[資訊] 索引標籤中，確認模型化閘道匯入的結果。

請依循下列步驟：

1. 在 [OneClick 主控台] 按一下適用的 VNM 模型。
2. 在 [元件詳細資料] 面板中，按一下 [資訊] 索引標籤。
3. 按一下 [模型化閘道] 以展開區段。
4. 檢閱 [模型化閘道] 區段中的表格，以取得最近匯入作業的相關資訊。此表格包含下列資訊：

匯入檔

顯示匯入檔的名稱。

記錄檔

顯示記錄檔的名稱。

開始時間

指出拓撲匯入程序何時開始。

結束時間

指出拓撲匯入程序何時完成。

進度

進度欄位顯示尚未完成的拓撲匯入作業的狀態。此欄位可能的值為：

- 初始化中
- 識別模型中
- 建立模型中
- 啓動模型中
- 對應連線中
- 放置模型中
- 建立連線中
- 更新模型中
- 終結模型中

- 完成
- 已中斷連線

錯誤

顯示匯入過程中產生的錯誤數。

建立的模型

顯示匯入過程中建立的模型數。

終結的模型

顯示匯入過程中刪除的模型數。

更新的模型

顯示匯入過程中更新的模型數。

建立的連線

顯示匯入過程中建立的連線數。

移除的連線

顯示匯入過程中移除的連線數。

5. 按一下 [最大記錄] 設定連結，以修改表格中列出的匯入檔數目。
6. 視需要按一下任何表格欄標頭來排序資料。
7. 在 [篩選] 欄位中輸入文字，以依特定的準則來限制匯入資料。
8. 按一下 [更新]，檢查正在處理的匯入作業的狀態。
畫面會重新整理並顯示最新的匯入資訊。

錯誤記錄

所有錯誤及可能的原因都記錄在錯誤記錄檔中。依預設，匯入工具會建立名為 *<nameofimportfile>.log* 的錯誤記錄，其中 *<nameofimportfile>* 是匯入檔的名稱。您也可以使用關於匯入工具的一節中所規定的語法，指定記錄檔的特定名稱。匯入完成時，記錄檔會出現在 *SS-Tools* 目錄中。記錄檔中會記錄成功建立、刪除及更新模型和連線的次數。記錄檔也會記錄匯入過程中發生的每一次失敗。

第 3 章：從 CA Spectrum 匯出拓撲資料

本節包含以下主題：

[關於從 CA Spectrum 匯出拓撲資料](#) (位於 p. 35)

[配置匯出設定](#) (位於 p. 36)

[ExportConfiguration 元素](#) (位於 p. 37)

[用於匯出的 modelingateway 工具](#) (位於 p. 39)

關於從 CA Spectrum 匯出拓撲資料

模型化閘道支援從 SpectroSERVER 匯出拓撲資訊和設定設定。資訊會匯出成 XML 格式的檔案，您可使用模型化閘道將此檔案匯入到指定的 SpectroSERVER。

依預設會匯出下列類型的資訊：

- Device 元素，含配置屬性。
- Port 元素，含配置屬性。
- Container 元素，含配置屬性。
- 連線 (已解析和未解析的 WA_Link 連線)。
- 通用拓撲階層。
- 通用拓撲中每一個檢視的設定，包括注釋和縮放資訊，但不包括背景影像。
- 使用者模型和整個使用者設定，例如使用者相關的關係、屬性，以及 LicenseRole、AccessGroup、PrivilegeRole 及 UserGroup 等模型。
- Discovery 配置。
- 服務管理設定和屬性。
- 靜態和動態全域集合，包括每一個全域集合中的所有模型、所有動態集合準則、縮放清單、分組清單，以及拓撲設定。

重要！ 轉換器工具的用途是將 SpectroSERVER 資料庫劃分到兩個以上的 SpectroSERVER，或變更單一 SpectroSERVER 資料庫的範圍控制代碼。如果您使用模型化閘道匯出功能的目的與轉換器工具一樣，請注意並不是所有資訊都會匯出。若要為匯出功能目前不支援的資料，配置行為和設定模型，則在匯入匯出的資料之後，必須手動執行一些工作。不會匯出的資料類型的例子包括 (但不限於) Service Performance Manager (SPM) 測試和 Policy Manager 原則。

配置匯出設定

您可以在 `.modelinggatewayresource.xml` 檔中修改 `ExportConfiguration` 元素，以控制要匯出的資訊。預設會匯出 [通用] 容器下的所有拓撲和模型化資訊。您可以修改 `RootContainerToExport` 元素，以指定要從中匯出的不同根容器。根容器及其每一個子容器的所有內容都會匯出。

附註： 您不需要使用 DTD 來匯出資料；DTD 只用於匯入資料。

針對裝置、容器及連接埠來匯出的屬性，分別定義在下列元素中：`DeviceExportAttributes`、`ContainerExportAttributes` 及 `PortExportAttributes`。請視需要在這些元素中新增和移除屬性。

當 `ExportConfiguration` 元素中的 `export_spectrum_settings` 旗標設定為 `true` 時，`SpectrumConfigurationExport` 元素控制要匯出何種 CA Spectrum 設定資料。

例如，下列元素控制針對 `LostFound` 模型來匯出的屬性：

```
<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>
```

更多資訊：

[ExportConfiguration 元素](#) (位於 p. 37)

ExportConfiguration 元素

若要控制匯出行爲，請在 `.modelinggatewayresource.xml` 檔中使用 ExportConfiguration 元素。

ExportConfiguration 元素有下列語法：

```
<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

匯出裝置模型。

export_containers

匯出容器模型。

export_port_attributes

匯出連接埠屬性。

export_links

匯出裝置連結。

export_topology_layout

匯出裝置和容器模型在拓撲中的 x,y 座標。

export_annotation

匯出注釋和模型群組資訊。

export_WA_Link_models

匯出 WA_Link 模型。如果您決定不匯出 WA_Link 模型，則系統會將模型視為透明。兩個裝置模型之間的廣域連結會匯出為直接連結。

export_spectrum_settings

匯出 CA Spectrum 設定，例如錯誤隔離、Discovery 及 VNM 控制的設定。

export_user_models

匯出使用者模型、使用者授權、使用者權限、使用者喜好設定，以及其他所有使用者相關的關係屬性和模型。

export_servicemodeling

匯出服務管理設定和屬性。

附註：如需 Service Manager 的詳細資訊，請參閱《*Service Manager 使用者指南*》。

export_schedules

匯出排程。

export_global_collections

匯出靜態和動態全域集合，包括每一個全域集合中的所有模型、所有動態集合準則、縮放清單、分組清單，以及拓撲設定。

export_discovery_configs

匯出 Discovery 配置。

export_from_primary_ss_only

指定當主要 SpectroSERVER 停機時，模型化閘道是否應連線至次要 SpectroSERVER。

export_policy_manager

匯出原則管理員原則。包括所有相關的模型、原則、規則、權限及範本。

範例：

下列範例會匯出所有資訊，但連接埠屬性資訊除外。此範例中也指示當主要 SpectroSERVER 停機時，模型化閘道不要連線至次要 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"
```

```
/>
```

用於匯出的 modelinggateway 工具

模型化閘道命令列工具 'modelinggateway' (在 Windows 中是 modelinggateway.bat) 位於 SS-Tools 目錄中。其匯出語法為：

Windows

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

Solaris/Linux

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

-vnm vnm_name

指定 SpectroSERVER 主機的名稱。

-cldb

(選用) 以整合 CA Spectrum 與 CA C MDB 時可用的格式來匯出 SpectroSERVER 的內容。如需實作此整合的詳細資訊，請連絡 CA 支援。

-e export_file

匯出 CA Spectrum 拓撲資料。

-o *outputfile*

(選用) 將錯誤資訊記錄到 *outputfile* 參數中指名的檔案。如果不使用此選項，則錯誤資訊會記錄到名為 *export_file.log* 的檔案。 *Export_file* 是 XML 檔的名稱。

-debug *debugfile*

(選用) 指出您想在匯出過程中建立偵錯輸出檔。使用 **-debug** 選項時，您可以提供自己的偵錯檔名進行輸出。如果未提供 *debugfile* 的值，則偵錯檔名預設為 *export_file* 名稱，結尾是 *.debug*。

附註：在執行模型化閘道的機器上，**-debug** 選項需要磁碟空間。*export_file* 中的模型數目會影響偵錯輸出檔的大小：資料庫中的模型數目越大，產生的偵錯檔就越大。

附註：若要在另一部伺服器上執行 modelinggateway 工具，請將 modelinggateway 工具及其所有支援檔案移至該伺服器。如需詳細資訊，請參閱《分散式 SpectroSERVER 管理指南》。

匯出 CA Spectrum 拓撲資料

使用 modelinggateway 工具來匯出 CA Spectrum 拓撲資料。

請依循下列步驟：

若要匯出 CA Spectrum 拓撲資料，請使用 **-e** 旗標。例如，執行下列命令會將資料從 NOC1_Spectrum 上的 SpectroSERVER，匯出到模型化閘道格式的 XML 檔，名為 NOC1_data.xml：

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

匯入模型化閘道 XML 檔

您可以將資料從模型化閘道格式的 XML 檔匯入到 CA Spectrum。

請依循下列步驟：

若要從模型化閘道格式的 XML 檔匯入，請使用 **-i** 旗標。例如，執行下列命令會將 NOC1_data.xml 中的資料，匯入到 NOC2_Spectrum 中的 SpectroSERVER。

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

附錄 A：文件類型定義元素

本節說明文件類型定義 (DTD) 中定義的每一個元素的功能。本節也提供每一個元素的內容。本節不說明 DTD 中使用的 XML 語法。如需語法資訊，請參閱 XML 參考資料。

關聯

語法

上層元素：

- 更新
- 終結

子件元素：

- Left_Model
- Right_Model

規則： Association 元素必須包含一個 Left_Model 元素和一個 Right_Model 元素。

使用量

Association 元素建立或終結模型之間的關聯。

如果將 Association 元素做為 Destroy 元素的下層元素，則會終結指定的關聯。如果將 Association 元素做為 Update 元素的下層元素，則會建立指定的關聯。

屬性

關係

指定此關聯中 Left_Model 與 Right_Mode 間之 CA Spectrum 關係的名稱或控制代碼。

連線

語法

上層元素：

- 拓撲
- Topology_Container
- 更新
- 終結

子件元素：Device

規則：Connection 元素必須包含兩個裝置元素。

使用量

Connection 元素指定兩個裝置之間的連線。Connection 元素必須一律包含兩個 Device 元素，而其中每一個 Device 元素可以包含零個或一個 Port 元素。如果指定一或多個連接埠，則會在連接埠層級解析連線。如果未指定連接埠，則會觸發 Discovery 來尋找連線的連接埠。

如果將 Connection 元素做為 Destroy 元素的下層元素，則會終結指定的連線。如果在其他任何內容下使用連線，則會建立連線。

屬性

create_pipe

指出是否在 OneClick 中顯示所指定連線的圖形表示法。

預設值：True

相關

語法

上層元素：Import

子件元素：Correlation_Domain

規則：Correlation 元素可以包含任何數量的下層元素。

使用量

Correlation 元素代表 Correlation Manager 模型，且與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

無。

Correlation_Domain

語法

上層元素：Correlation

子件元素：

- 裝置
- 連接埠
- Model_Attr
- GenericView_Container

規則：Correlation_Domain 元素可以包含任何數量的下層元素。

使用量

Correlation_Domain 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A

CustomerManager

語法

上層元素：SM_Service_Mgt

子件元素：

- SM_Customer
- SM_CustomerGroup

規則：CustomerManager 元素可以包含任何數量的這些下層元素。

使用量

CustomerManager 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

附註：如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
containment_relation	字元	Groups_Customers	N/A
model_type	字元	CustomerManager	N/A

終結

語法

上層元素：Import

子件元素：

- Topology_Container
- Location_Container

- GenericView_Container
- 裝置
- 模型
- 連線
- EventModel
- SM_Service
- SM_AttrMonitor
- SM_LatencyMon
- SM_ConnectMon
- SM_SLA
- SM_Guarantee
- SM_Customer
- 關聯

規則： Destroy 元素可以視需要包含任何數量的上述每一個下層元素。

使用量

使用 Destroy 元素來移除容器模型、裝置模型、連線及關聯。您不能在 Destroy 元素中鑲嵌元素來表達階層或終結階層。Destroy 元素中唯一允許的階層是 Connection-Device-Port 階層，其在連接埠層級指定要終結的連線。您可以終結容器模型，而不終結其內含的裝置模型或其他容器模型。在此情況下，剩餘的模型會放入 CA Spectrum「尋獲失物」中。

屬性

Destroy 元素沒有任何屬性。

裝置

語法

上層元素：

- 拓撲
- 位置

- Topology_Container
- Left_Model
- Right_Model
- Location_Container
- GenericView
- GenericView_Container
- 連線
- 更新
- 終結
- SM_Service
- SM_AttrMonitor

子件元素：

- 連接埠
- 排程

規則：

- **Port:**如果 Device 元素放在 Connection 元素內，則只允許一個 Port 元素。如果 Device 元素放在 Update 元素內來更新連接埠，則允許多個 Port 元素。如果 Device 元素放在 View、Container 或 Destroy 元素內，則會忽略連接埠。
- **Schedule：**Device 元素可以包含一個 Schedule 元素。

使用量

若要建立、終結或更新裝置模型，請使用 Device 元素。Device 元素可讓您使用 IP 位址或 DNS 名稱來定義裝置模型。

附註：如果建立裝置時未提供 community_string 和 agent_port 屬性，CA Spectrum 會使用預先定義的 SNMP 憑證來建立裝置。這些憑證是在 OneClick 中的 VNM 模型 [資訊] 索引標籤 > [自動搜索控制] 子檢視 > [模型化和通訊協定選項] 區段中配置。

屬性

ip_dnsname

指定裝置的 IP 位址或 DNS 名稱。如果裝置不支援 SNMP 通訊，您可以在這裡使用唯一字串，並指定 model_type。

secdomain_ipname

(選用) 指定主機的 IP 位址，此主機在裝置所在的安全網域中執行 SDConnector。

預設值：0.0.0.0

model_handle

(選用) 指定 model_handle 來識別現有的裝置模型。

附註：如果提供 model_handle，則會忽略 ip_dnsname 的值。

model_type

(選用) 用來模型化裝置的 CA Spectrum 模型類型。此裝置模型可以是 .modelinggatewayresource.xml 檔中定義的任何智慧模型類型。

附註：如果已提供有效的 IP 位址或 DNS 名稱，則這裡不需要指定值。

community_string

(選用) 裝置的通訊字串。

附註：如果不加上 community_string，CA Spectrum 會使用第一個 SNMP 社群字串值來建立裝置。這些值是在 OneClick 中的 VNM 模型 [資訊] 索引標籤 > [自動搜索控制] 子檢視 > [模型化和通訊協定選項] 子檢視中指定。

agent_port

(選用) 控制與裝置的 SNMP 代理程式通訊時使用的連接埠號碼。

附註：如果不加上 agent_port，CA Spectrum 會使用第一個 SNMP 連接埠值來建立裝置。這些值是在 OneClick 中的 VNM 模型 [資訊] 索引標籤 > [自動搜索控制] 子檢視 > [模型化和通訊協定選項] 子檢視中指定。

is_managed

(選用) 設定為 true 時，讓裝置模型進入維護模式。

預設值：True

poll_interval

(選用) 指定 SpectroSERVER 讀取裝置模型中所有已標記為 POLLED 的屬性時，要相距的時間間隔 (以秒為單位)。

log_ration

(選用) 指定 SpectroSERVER 在將輪詢結果記錄到資料庫之前，要輪詢裝置的次數。

poll_status

(選用) 可讓您將輪詢狀態設為 false，以停用 SpectroSERVER 進行的裝置輪詢。

model_name

(選用) 指定模型的名稱。

DeviceType

(選用) 指定模型的裝置類型。

附註：如需裝置類型的詳細資訊，請參閱《憑證使用者指南》。

reconfig

(選用) 指定模型化閘道是否傳送動作給 SpectroSERVER 來重新配置裝置模型。

discover_connections

(選用) 設定為 true 時，在任何新建立的裝置模型上執行 Discovery，以自動對應模型連線。

EventModel

語法

上層元素：

- Topology_Container
- Left_Model
- Right_Model

子件元素：無

規則：N/A

使用量

若要匯入 EventModel 模型以用於 Southbound Gateway 整合，請使用 EventModel 元素。如需 Southbound Gateway 的詳細資訊，請參閱《*Southbound Gateway 工具組指南*》。

屬性

model_name

指定所具現化或識別之模型的唯一名稱。

unique_id

指定識別碼，用來唯一定義此 EventModel 模型所代表的事件來源。如需詳細資訊，請參閱《*Southbound Gateway 工具組指南*》。

model_handle

(選用) 指定 model_handle 來識別現有的裝置模型。

Security_String

(選用) 指定 EventModel 的安全性字串。

預設值：public

manager_name

(選用) 指定正在使用 Southbound Gateway 的第三方應用程式的名稱。

附註：對於這裡未列出的任何應用程式，請使用預設值。

預設值：0

1

NetMentor

2

SSM

3

Omni2000

GenericView

語法

上層元素：Import

子件元素：

- GenericView_Container
- 裝置

規則：GenericView 元素可以包含任何數量的這些下層元素。

使用量

若要建立拓撲檢視和位置檢視以外的自訂階層式檢視，請使用 GenericView 元素。您可以修改此元素以符合您的整合需求。

屬性

containment_relation

指定關係控制代碼，此關係控制代碼定義哪一個 CA Spectrum 關係定義了此檢視內的包含關係。

限制：必須是 CA Spectrum 包含關係。

model_type

指定模型類型，代表定義給此檢視的最上層容器模型。此 model_type 必須和模型控制代碼一起指定在 .modelinggatewayresource.xml 檔中。

名稱

指定 GenericView 階層中所具現化的最高容器模型的唯一名稱。

complete_topology

(選用) 設定為 true 時，終結 GenericView 檢視中任何未指定、現有的容器和裝置模型。同時終結該檢視之子容器中的這些模型。

GenericView_Container

語法

上層元素：GenericView

子件元素：

- GenericView_Container
- 裝置

規則：GenericView_Container 元素可以包含任何數量的下層元素。

使用量

若要在一般檢視中建立容器模型，請使用 GenericView_Container 元素。GenericView 和 GenericView_Container 元素都可用來建立自訂的檢視。因此，身為整合者的您，應該決定何時或如何使用此容器。

屬性

名稱

指定所具現化或識別之模型的名稱。需要 `model_type` 和 `name` 屬性才能唯一識別 GenericView_Container。依預設，此屬性用來設定 CA Spectrum 模型名稱屬性 (屬性 ID 0x1006e) 的值。不過，在 `.modelinggatewayresource.xml` 檔中，此屬性可以變更為其他任何屬性。您可以變更 `.modelinggatewayresource.xml`，讓 `name` 對應到不同屬性。在此情況下，該新屬性 (以及模型類型) 會用來識別容器。此行為讓兩個容器有相同的模型名稱。

`model_type`

指定用來建立模型的 CA Spectrum 模型類型。此 `model_type` 必須和模型控制代碼一起指定在 `.modelinggatewayresource.xml` 檔中。需要 `model_type` 和 `name` 屬性才能唯一識別 GenericView_Container。

`containment_relation`

(選用) 在 Generic_Container 與該容器內的模型之間，存在的 CA Spectrum 關係的名稱。如果未指定此屬性的值，則會繼承上層模型的包含關係。

GlobalCollection

語法

上層元素：Import

子件元素：

- 裝置
- Topology_Container
- Location_Container

使用量

代表 GlobalCollection 模型。

屬性

名稱

指定此全域集合的名稱。

containment_relation

指定關係控制代碼，此關係控制代碼定義哪一個 CA Spectrum 關係定義了此檢視內的包含關係。

預設值："GlobalCollect"

collectionDescription

(選用) 說明全域集合。

Security_String

(選用) 指定全域集合的安全性字串。

匯入

語法

上層元素：無

子件元素：

- 拓撲
- 位置
- GenericView
- 更新
- 終結
- SM_Service_Mgt
- 相關
- GlobalCollection

規則：Import 元素可以包含上述每一個下層元素的其中一個。

使用量

Import 元素是根元素，必須出現在每一個輸入檔中。

屬性

model_activation_time

指定每一次啟動裝置模型時允許的最長時間 (分鐘)。

資料類型：字元

預設值：5 分鐘

Left_Model

語法

上層元素：Association

子件元素：

- 裝置
- 連接埠
- Topology_Container
- Location_Container
- EventModel
- 模型

規則：Left_Model 元素只能包含一個下層元素。

使用量

Left_Model 元素定義位於關聯左側的模型。

屬性

無。

List_Value

語法

上層元素：Model_Attr

子件元素：無

規則：N/A

使用量

若要指定 CA Spectrum 清單屬性值，請使用 List_Value 元素。

位置

屬性

無。

語法

上層元素：Import

子件元素：

- Location_Container
- 裝置

規則：Location 元素可以包含任何數量的下層元素。

使用量

若要指定想要在 OneClick 的位置檢視 (World 拓撲) 中建立模型，請使用 Location 元素。

屬性

complete_topology

(選用) 設定為 true 時，匯入期間會終結位置檢視或任何子容器中的任何未指定、現有的容器和裝置模型。

預設值：False

資料類型：布林值

Location_Container

語法

上層元素：

- 位置
- Location_Container
- Left_Model

- Right_Model
- GlobalCollection

子件元素：

- Location_Container
- 裝置

規則：Location_Container 元素可以包含任何數量的下層元素。

使用量

若要建立或指定 Location_Container 模型，以便於位置檢視中將模型和其他位置容器分組，請使用 Location_Container 元素。

屬性

名稱

所具現化或識別之模型的名稱。需要 model_type 和 name 屬性才能唯一識別 Location_Container。

依預設，此屬性用來設定 CA Spectrum 模型名稱屬性 (屬性 ID 0x1006e) 的值。不過，在 .modelinggatewayresource.xml 檔中，此屬性可以變更為其他任何屬性。您可以變更 .modelinggatewayresource.xml，讓 name 對應到不同屬性。在此情況下，該新屬性 (以及模型類型) 會用來識別容器。此行為讓兩個容器有相同的模型名稱。

model_type

指出您要建立的模型類型。需要 model_type 和 name 屬性才能唯一識別 Location_Container。可能值包括：

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

model_handle

(選用) 可用來識別模型。如果提供 `model_handle`，則會忽略 `name` 和 `model_type` 的值。

Security_String

(選用) 定義 CA Spectrum 使用者存取模型時的必要條件。每一個安全性字串包含一或多個安全性社群項目，並已指派給模型。

model_name

(選用) 若要變更模型的名稱，請使用 `name` 和 `model_name` 屬性。`name` 屬性指定舊名稱，`model_name` 屬性指定新名稱。

model_modify_author

(選用) 將資料寫入 CA Spectrum 屬性 `mdl_modify_attr`。

complete_topology

(選用) 設定為 `true` 時，匯入期間會終結位置檢視或任何子容器中的任何未指定、現有的容器和裝置模型。

預設值：False

Model_Attr

語法

上層元素：Correlation_Domain

子件元素：List_Value

規則：Model_Attr 元素可以包含任何數量的下層元素。

使用量

若要指定以多行文字或值清單做為值的 CA Spectrum 屬性，請使用 Model_Attr 元素。

屬性

attr_id

指出您指定的屬性的 CA Spectrum 屬性 ID。

資料類型：字元

模型

語法

上層元素：

- Left_Model
- Right_Model

子件元素：無

使用量

若要代表任何 CA Spectrum 模型，請使用 Model 元素。

屬性

名稱

指定此模型的名稱。

model_type

指定此模型的模型類型。

model_handle

(選用) 指定此模型的 model_handle。如果指定 model_handle 值，則會忽略 name 和 model_type 值。

MonitorPolicy_Attr

語法

上層元素：

- SM_Service
- SM_AttrMonitor

子件元素：無

規則：N/A

使用量

MonitorPolicy_Attr 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《Service Manager 使用者指南》。

屬性

無。

連接埠

語法

上層元素：

- 裝置
- Left_Model
- Right_Model
- Correlation_Domain
- SM_Service
- SM_AttrMonitor

子件元素：無

規則：N/A

使用量

Port 元素用於在連接埠層級指定連線，或更新連接埠屬性。更新時，上層 Device 元素放在 Update 元素內。指定連線時，上層 Device 元素放在 Connection 元素內。

屬性

identifier_name

搭配 identifier_value 屬性來唯一識別連接埠。identifier_name 可以是「可能值」欄中列出的任何 MIB OID 名稱。portID 值可用來依 Component_OID 屬性 (0x1006a) 來識別連接埠。如果 Port 元素代表框架轉送虛擬電路，請使用 frCircuitTableInstance。如果 Port 元素代表 ATM 虛擬通道或路徑連結，請使用 atmVclTableInstance。

portID 值可用來依 Component_OID 屬性 (0x1006a) 來識別連接埠。如果 Port 元素代表框架轉送虛擬電路，請使用 frCircuitTableInstance。如果 Port 元素代表 ATM 虛擬通道或路徑連結，請使用 atmVclTableInstance。可能值包括：

- ifIndex
- ipAddress
- ifPhysAddress
- ifName
- ifAlias
- model_name
- portDescription
- portID
- frCircuitTableInstance
- atmVclTableInstance
- atmVplTableInstance

identifier_value

指定 identifier_name 選取項目的值。

model_handle

(選用) 指定 model_handle 來識別現有的模型。

附註：如果提供 model_handle，則會忽略 identifier_name 和 identifier_value 的值。

ip_dnsname

(選用) 指定連接埠模型的 IP 位址或 DNS 名稱。如果連接埠模型不支援 SNMP 通訊，您可以在這裡使用唯一字串，並指定 model_type。

model_name

(選用) 指定您要更新的模型的名稱。

circuit_id

(選用) 依 ID 來識別 ATM 或框架轉送連線中的相關電路。

circuit_name

(選用) 依名稱來識別 ATM 或框架轉送連線中的相關電路。

log_ratio

(選用) 指定在將輪詢結果記錄到資料庫之前，發生連接埠模型輪詢的次數。

poll_interval

(選用) 指定 SpectroSERVER 讀取已標記為 POLLED 的連接埠模型中的所有屬性時，要相距的指定時間間隔 (以秒為單位)。

poll_status

(選用) 讓管理員將輪詢狀態設為 **False**，以停用連接埠模型輪詢。

Right_Model

語法

上層元素：Association

子件元素：

- 裝置
- 連接埠
- Topology_Container
- Location_Container
- EventModel
- 模型

規則：Right_Model 元素只能包含一個下層元素。

使用量

Right_Model 元素定義位於關聯右側的模型。

屬性

無。

RTM_Test

語法

上層元素：

- SM_Service
- SM_AttrMonitor

子件元素：無

規則：N/A

使用量

RTM_Test 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《Service Manager 使用者指南》。

屬性

如需屬性定義，請參閱《Service Manager 使用者指南》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
model_type	字元	RTM_Test	N/A

排程

語法

上層元素：Device

子件元素：無

使用量

若要建立特定裝置模型的維護模式排程，請使用 Schedule 元素。

屬性

名稱

指定排程的名稱。

SCHED_Recurrence

指定裝置模型進入維護模式的頻率。

1

永遠 (24x7)

2

每日

3

每週

4

每月

5

每年

預設：1

SCHED_Start_Hour

(選用) 指定裝置進入維護模式的小時。

限制：0-23

SCHED_Start_Minute

(選用) 指定裝置進入維護模式的分鐘。

限制：0-59

SCHED_Start_DoW

(選用) 指定裝置在星期幾進入維護模式。

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

(選用) 指定裝置在每月幾號進入維護模式。

限制：1-31

SCHED_Start_Month

(選用) 指定裝置進入維護模式的月份。

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

(選用) 指定裝置處於維護模式的時間長度 (以秒為單位)。

預設值：0

SCHED_Recurrence_Multiplier

(選用) 指定週期單位數 (日、週、月、年)，決定每次排定的維護模式開始之間相隔的時間長度。

預設：1

SCHED_Daily_Repeat_Limit

(選用) 指定在每個週期內要連續重複所排定之維護 (以 SCHED_Start_Hour 和 SCHED_Start_Minute 指定) 的天數。此屬性僅適用於每週、每月或每年週期。

SM_AttrMonitor

語法

上層元素：SM_Service

子件元素：無

規則：N/A

使用量

SM_AttrMonitor 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
containment_relation	字元	N/A	SLMMonitors SLMWatchesContainer
AttrToWatch	字元	N/A	N/A
MonitorPolicy_ID	字元	N/A	N/A
is_managed	布林值	N/A	True False
Generate_Service_Alarms	布林值	N/A	True False
model_type	字元	SM_AttrMonitor	N/A

SM_Customer

語法

上層元素：

- SM_CustomerGroup
- CustomerManager

子件元素：無

規則：N/A

使用量

SM_Customer 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
containment_relation	字元	N/A	SlmAgreesTo SlmUses
Security_String	字元	N/A	N/A
CustomerID	字元	N/A	N/A
Criticality	字元	N/A	N/A
CustomerField4	字元	N/A	N/A
CustomerField5	字元	N/A	N/A
CustomerField6	字元	N/A	N/A

屬性	資料類型	預設值	可能值
CustomerField7	字元	N/A	N/A
Contact_Name	字元	N/A	N/A
Contact_Title	字元	N/A	N/A
Contact_Location	字元	N/A	N/A
Email_Address	字元	N/A	N/A
Phone_Number	字元	N/A	N/A
Mobile_Phone_Number	字元	N/A	N/A
Pager_Number	字元	N/A	N/A
Fax_Number	字元	N/A	N/A
User_Defined_1	字元	N/A	N/A
User_Defined_2	字元	N/A	N/A
User_Defined_3	字元	N/A	N/A
User_Defined_4	字元	N/A	N/A
Secondary_Contact_Name	字元	N/A	N/A
Secondary_Contact_Location	字元	N/A	N/A
Secondary_Email_Address	字元	N/A	N/A
Secondary_Phone_Number	字元	N/A	N/A
Secondary_Mobile_Phone_Number	字元	N/A	N/A

屬性	資料類型	預設值	可能值
Secondary_Pager_Number	字元	N/A	N/A
Secondary_Fax_Number	字元	N/A	N/A
Secondary_User_Defined_1	字元	N/A	N/A
Secondary_User_Defined_2	字元	N/A	N/A
Secondary_User_Defined_3	字元	N/A	N/A
Secondary_User_Defined_4	字元	N/A	N/A
model_type	字元	SM_Customer	N/A

SM_CustomerGroup

語法

上層元素：

- CustomerManager
- SM_CustomerGroup

子件元素：

- SM_CustomerGroup
- SM_Customer

規則：SM_CustomerGroup 元素可以包含任何數量的這些下層元素。

使用量

SM_CustomerGroup 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	NA	N/A
containment_relation	字元	Groups_Customer	N/A
model_type	字元	SM_CustomerGroup	N/A

SM_Guarantee

語法

上層元素：SM_SLA

子件元素：無

規則：N/A

使用量

SM_Guarantee 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
containment_relation	字元	SlmsMeasuredBy	N/A

屬性	資料類型	預設值	可能值
is_managed	布林值	N/A	True False
DegradedTimeViolationLevel	字元	N/A	N/A
DegradedTimeWarningLevel	字元	N/A	N/A
DownTimeViolationLevel	字元	N/A	N/A
DownTimeWarningLevel	字元	N/A	N/A
LorTimeViolationLevel	字元	N/A	N/A
LorTimeWarningLevel	字元	N/A	N/A
model_type	字元	SM_Guarantee	N/A

SM_LatencyMon

語法

上層元素：

- SM_Guarantee
- SM_AttrMonitor
- SM_Service

子件元素：

- Topology_Container
- MonitorPolicy_Attr

規則：SM_LatencyMon 元素可以包含任何數量的這些下層元素。

使用量

SM_LatencyMon 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《Service Manager 使用者指南》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
containment_relation	字元	N/A	SlmMonitors SlmWatchesContainer
is_managed	布林值	N/A	True False
DefaultMaxRTT	字元	N/A	N/A
DefaultMeasureInterval	字元	N/A	N/A
mode_type	字元	SM_LatencyMon	N/A

SM_Service

語法

上層元素：

- SM_Service
- SM_ServiceMgr

子件元素：

- SM_Service
- SM_AttrMonitor

規則：SM_Service 元素可以包含任何數量的這些下層元素。

使用量

SM_Service 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
Criticality	字元	N/A	N/A
containment_relation	字元	N/A	SlmMonitors SlmWatchesContainer
AttrToWatch	字元	N/A	N/A
MonitorPolicy_ID	字元	N/A	N/A
is_managed	布林值	N/A	True False
Generate_Service_Alarms	布林值	N/A	True False
Security_String	字元	N/A	N/A
model_type	字元	SM_Service	N/A

SM_Service_Mgt

語法

上層元素：Import

子件元素：

- SM_ServiceMgr
- CustomerManager
- SM_SLA_Mgr

規則：在 SM_Service_Mgt 中，上述每一個下層元素皆只能有一個執行個體存在。

使用量

SM_Service_Mgt 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	Service Management	N/A
containment_relation	字元		N/A
model_type	字元		N/A

SM_ServiceMgr

語法

上層元素：SM_Service_Mgt

子件元素：SM_Service

規則：SM_ServiceMgr 元素可以包含任何數量的這個下層元素。

使用量

SM_ServiceMgr 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱	字元	N/A	N/A
containment_relation	字元	SlmContains	N/A
model_type	字元	SM_ServiceMgr	N/A

SM_SLA

語法

上層元素：SM_SLA_Mgr

子件元素：SM_Guarantee

規則：SM_SLA 元素可以包含任何數量的這個下層元素。

使用量

SM_SLA 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱 (必要)	字元	N/A	N/A
containment_relation	字元	SlmContains	N/A
is_managed	布林值	N/A	True False
Security_String	字元	N/A	N/A

屬性	資料類型	預設值	可能值
model_type	字元	SM_SLA	N/A

SM_SLA_Mgr

語法

上層元素：SM_Service_Mgt

子件元素：SM_SLA

規則：SM_SLA_Mgr 元素可以包含任何數量的這個下層元素。

使用量

SM_SLA_Mgr 元素與 CA Spectrum Service Manager 搭配使用。如需用法詳細資料，請參閱《*Service Manager 使用者指南*》。

屬性

如需屬性定義，請參閱《*Service Manager 使用者指南*》。

屬性	資料類型	預設值	可能值
名稱	字元		N/A
containment_relation	字元	SlmContains	N/A
model_type	字元	SM_ServiceMgr	N/A

拓撲

語法

上層元素：Import

子件元素：

- Topology_Container
- 裝置
- 連線

規則：Topology 元素可以包含任何數量的這些下層元素。

使用量

若要在 OneClick 拓撲檢視 (通用拓撲) 中建立模型，請使用 Topology 元素。

屬性

complete_topology

設定為 true 時，在匯入期間終結拓撲檢視中任何未指定、現有的容器和裝置模型。同時終結該檢視的任何子容器。

預設值：False

discover_connections

設定為 true 時，Discovery 會在任何新建立的裝置模型上執行，以自動對應模型的連線。

預設值：False

Topology_Container

語法

上層元素：

- 拓撲
- Topology_Container
- SM_Service
- SM_AttrMonitor

子件元素：

- Topology_Container
- 裝置
- EventModel
- 連線

規則：Topology_Container 元素可以包含任何數量的這些下層元素。

使用量

若要建立或指定 Topology_Container 模型，以便於拓撲檢視中將模型和其他拓撲容器分組，請使用 Topology_Container 元素。以下 model_type 一節中的表格列出可能的模型類型。

屬性

model_type

指出您要建立的模型類型。需要 model_type 和 name 屬性才能唯一識別 Topology_Container。

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

model_handle

(選用) 可用來識別現有的模型。如果提供 model_handle，則會忽略 model_type 和 model_name 的值。

Security_String

(選用) 指定指派給模型的 CA Spectrum 安全性層級。

subnet_address

(選用) 指定裝置的子網路位址。

subnet_mask

(選用) 指定遮罩，以決定裝置 IP 位址所屬的子網路。

model_name

(選用) 指定容器模型的模型名稱。

trapIPAddress

(選用) 僅適用於 EventAdmin 模型。

x_coordinate

(選用) 指定模型在拓撲中的 x 座標。

y_coordinate

(選用) 指定模型在拓撲中的 y 座標。

complete_topology

(選用) 設定為 True 時，匯入期間會終結此 Topology_Container 中的任何未指定、現有的容器和裝置模型，以及任何子容器。

預設值：False

discover_connections

(選用) 指定 CA Spectrum 是否搜索和模型化已連接到此模型的裝置。

更新

語法

上層元素：Import

子件元素：

- Topology_Container
- Location_Container
- GenericView_Container
- 連線
- 裝置

- 模型
- EventModel
- SM_Service
- SM_AttrMonitor
- SM_LatencyMon
- SM_ConnectMon
- SM_SLA
- SM_Guarantee
- SM_Customer
- 關聯

規則：Update 元素可以包含任何數量的這些下層元素

使用量

若要更新任何裝置、容器或連接埠子元素的屬性，請使用 Update 元素。

附註：使用此元素時，不能進行階層式指定，但在 Device 元素內使用 Port 元素除外。

屬性

無。

附錄 B：文件類型定義檔

本節包含文件類型定義 (DTD)，其中定義要匯入的 XML 元素和屬性。

附註：下列程式碼可能不是此檔案的最新版本。如需 DTD 的最新版本，請使用模型化閘道工具組實際隨附的檔案。此檔案位於 SS-Tools 目錄中，名稱爲 .modelinggateway.dtd。

```
<!-- ***** -->
<!-- Import 根元素包含 0 個或 1 個 Topology、 -->
<!-- Location、GenericView、Update、Destroy 元素、 -->
<!-- SM_Service_Mgt、Correlation 及全域集合。 -->
<!-- -->
<!-- 此元素有一個屬性 model_activation_time，表示 -->
<!-- 每一次啓動裝置模型的最長等待時間-->
<!-- (分鐘)。預設爲 5 分鐘。 -->
<!-- ***** -->

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

<!ATTLIST Import model_activation_time CDATA "5">

<!-- ***** -->
<!-- Topology 元素 (用於拓撲 -->
<!-- 檢視) 包含任何數量的 Topology_Container、 -->
<!-- Device 及 Connection 元素。 -->
<!-- -->
<!-- 此元素有兩個屬性：complete_topology -->
<!-- 和 discover_connection。 -->
<!-- -->
<!-- ***** -->

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

```

<!-- ***** -->
<!-- Topology_Container 元素 (用於 -->
<!-- 拓撲容器模型) 包含任何數量的 -->
<!-- Topology_Container、Device 及 Connection 元素。 -->
<!-- -->
<!-- 需要 "model_type" 和 "name" 屬性 -->
<!-- 才能唯一識別 Topology_Container 模型。 -->
<!-- -->
<!-- "model_handle" 也可用來識別模型。 -->
<!-- 如果提供 "model_handle", 則會忽略 "name" -->
<!-- 和 "model_type" 的值。 -->
<!-- -->
<!-- "trapIPAddress" 屬性僅應用於 -->
<!-- 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 >

```

```

<!-- ***** -->
<!-- Location 元素 (用於位置檢視) -->
<!-- 包含任何數量的 Location_Container 和 -->
<!-- Device 元素 -->
<!-- -->
<!-- 此元素具有屬性 complete_topology。 -->
<!-- ***** -->

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

<!-- ***** -->
<!-- Location_Container 元素 (用於 -->
<!-- Location 容器) 可以包含任何數量的 -->
<!-- Location_Container 和 Device 元素。 -->
<!-- -->
<!-- 需要 "model_type" 和 "name" 屬性 -->
<!-- 才能唯一識別 Location_Container 模型。 -->
<!-- -->
<!-- "model_handle" 也可用來識別模型。 -->
<!-- 如果提供 "model_handle", 則會忽略 "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 元素用於裝置模型。 -->
<!-- -->

```

```

<!-- 需要 "ip_dnsname" 屬性才能唯一 -->
<!-- 識別裝置模型。 -->
<!-- -->
<!-- model_handle" 也可用來識別裝置模型。 -->
<!-- 如果提供 "model_handle", 則會忽略 "ip_dnsname" -->
<!-- 的值。 -->
<!-- -->
<!-- 注意: -->
<!-- -->
<!-- 1. 如果 is_managed 屬性設定為 false, -->
<!-- 則無法聯繫裝置。 因此, 必須設定 model_type-->
<!-- 屬性。 -->
<!-- -->
<!-- 2. 在下列情況, 裝置可以包含 0 個、1 個 -->
<!-- 或 1 個以上的連接埠, 這些情況分別是: -->
<!-- -->
<!-- (a) 如果裝置放在 Container 或 Destroy 元素中, -->
<!-- 則不需要 Port 元素。 如果提供 Port -->
<!-- 元素, 則會忽略。 -->
<!-- -->
<!-- (b) 如果裝置放在 Connection 元素中, 則只能有 -->
<!-- 一個 Port 元素。 -->
<!-- -->
<!-- (c) 如果裝置放在 Update 元素中來更新 -->
<!-- 連接埠, 則可以有一個 Port 元素。 -->
<!-- -->
<!-- (d) 如果在裝置標籤上指定 discover_connections="true", -->
<!-- 請勿同時在該裝置的上層容器中做出相同的 -->
<!-- 指定。 這樣會導致效能和效率的相關 -->
<!-- 問題, 因為在容器上指定該屬性 -->
<!-- 會讓 Spectrum 在該容器中的每一個模型上 -->
<!-- 搜索連線。 -->
<!-- -->
<!-- ***** -->
<!-- ***** -->

<!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 元素用於裝置連接埠模型。 -->
<!-- -->
<!-- 需要 "identifier_name" 和 "identifier_value" -->
<!-- 屬性能唯一識別連接埠模型。 -->
<!-- -->
<!-- "model_handle" 也可用來識別連接埠模型。 -->
<!-- 如果提供 "model_handle"，則會忽略 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 >

```

```
<!-- ***** -->
<!-- 代表 Schedule 模型 -->
<!-- -->
<!-- SCHED_Recurrence 可以有下例值 -->
<!-- -->
<!-- 1 = 永遠 (24 x 7) -->
<!-- 2 = 每日 -->
<!-- 3 = 每週 -->
<!-- 4 = 每月 -->
<!-- 5 = 每年 -->
<!-- 6 = 一次 -->
<!-- -->
<!-- SCHED_Start_Hour : 值範圍 0-23 -->
<!-- SCHED_Start_Minute : 值範圍 0-59 -->
<!-- SCHED_Start_DoW : 星期幾 (範例 0-6, 星期日為 0), 適用於 -->
<!-- 每週週期 -->
<!-- SCHED_Start_DoM : 每月幾號 (範圍 1-31), 適用於每月和每年 -->
<!-- 週期 -->
<!-- SCHED_Start_Month : 範圍 0-11, 一月為 0, 適用於每年 -->
<!-- 週期 -->
<!-- SCHED_Duration : 使用中期間 (以秒為單位)。 可以為 0 (預設值) -->
<!-- SCHED_Recurrence_Multiplier : 週期單位數, -->
<!-- 決定每個使用中期間開始之間相隔 -->
<!-- 的時間長度。 -->
<!-- 預設值為 1。 -->
<!-- SCHED_Daily_Repeat_Limit : 在每個週期開始, -->
<!-- 連續重複每日排程 (以 -->
<!-- SCHED_Start_Hour 和 SCHED_Start_Minute 指定) -->
<!-- 的天數。 -->
<!-- 僅適用於每週、每月或 -->
<!-- 每年週期。 -->
<!-- SCHED_DayBitMask : 一週中「每週」排程應該啟動 -->
<!-- 的日子。 值包括: -->
<!-- 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, 如果想要週一、週三及週五, 則值等於 -->
<!-- 2+8+32=42 -->
<!-- SCHED_Start_MoY、 -->
<!-- SCHED_START_YEAR、 -->
```

```

<!-- SCHED_START_DAY：與 SCHED_START_MONTH 搭配使用，      -->
<!--          表示排程應該在未來某一天      -->
<!--          生效。 SCHED_START_YEAR 與      -->
<!--          SCHED_START_DAY 都不能為零。      -->
<!--          否則，排程會正常執行，      -->
<!--          並提早在今天生效。      -->
<!--          請注意，SCHED_START_YEAR 應該指定為      -->
<!--          自 1900 以來的年數。      -->
<!-- SCHED_Description：此排程的說明。      -->
<!--          *****      -->
<!--          *****      -->

<!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
>
<!-- ***** -->
<!-- 用來代表 EventModel 的元素。      -->
<!--          *****      -->
<!--          基於效能理由，必須指定唯一 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 元素代表裝置連線。 -->
<!-- 它包含連線中的兩個相關 Device 元素。 -->
<!-- 每一個 Device 可以有 0 個或 1 個 Port 元素。 -->
<!-- -->
<!-- Connection 具有一個屬性 create_pipe。對於 ATM -->
<!-- 電路連結，create_pipe 通常設定為 false，以避免 -->
<!-- 在檢視內建立太多管道。在此情況下， -->
<!-- 請使用 ATM Manager 來檢視連線。使用者 -->
<!-- 可決定 create_pipe 的設定。依預設， -->
<!-- 每一個連線會建立一個管道。 -->
<!-- -->
<!-- ***** -->

<!ELEMENT Connection (Device, Device)>

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

<!-- ***** -->
<!-- Update 元素用來更新 SPECTRUM 模型屬性和 -->
<!-- 關聯。 -->
<!-- -->
<!-- Update 元素可以包含任何數量要更新的 -->
<!-- Container、Device 及 Association 元素。 -->
<!-- 若要更新連接埠，必須將 Port 元素放入 -->
<!-- Device 元素內，再將 Device 元素放入 -->
<!-- Update 元素內。 -->
<!-- ***** -->
```

```
<!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 |
                    關聯
                    )* ) >

<!-- ***** -->
<!-- Destroy 元素：終結 SPECTRUM 模型和關聯。 -->
<!-- -->
<!-- Destroy 元素可以包含任何數量 -->
<!-- 要終結的 Container、Device、Connection 及 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 |
                    關聯
                    )* ) >
```

```
<!-- ***** -->
<!-- Association 元素定義兩個模型之間要建立或終結 -->
<!-- 的 SPECTRUM 關聯。 -->
<!-- -->
<!-- Association 元素包含一個 Left_Model 和一個 -->
<!-- Right_Model 元素。 -->
<!-- ***** -->

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

<!-- ***** -->
<!-- Left_Model 元素定義位於 Spectrum 關聯左側的 -->
<!-- 關聯。 -->
<!-- -->
<!-- Left_Model 元素只能包含一個下層 -->
<!-- 元素。 -->
<!-- ***** -->

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

<!-- ***** -->
<!-- Right_Model 元素定義位於 Spectrum 關聯右側的 -->
<!-- 關聯。 -->
<!-- -->
<!-- Right_Model 元素只能包含一個下層 -->
<!-- 元素。 -->
<!-- ***** -->

<!ELEMENT Right_Model (Device |
    Port |
    Topology_Container |
    Location_Container |
    EventModel |
    模型
) >

<!-- ***** -->
<!-- Model 元素可用來代表任何 SPECTRUM 模型。 -->
<!-- -->
<!-- 必須提供 model_type 和 name 才能定義模型。 -->
<!-- 應該使用 "model_type" 和 "name" 才能唯一識別 -->
```

```

<!-- 模型。 不過，如果提供 "model_handle"，-->
<!-- 則不會使用 "model_type" 和 "name" 的值。 -->
<!-- ***** -->

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

<!-- ***** -->
<!-- GenericView 元素（用於自訂的檢視） -->
<!-- 可以包含任何數量的 GenericView_Container 和 Device -->
<!-- 元素。 -->
<!-- -->
<!-- 此元素有 3 個必要屬性：containment_relation、-->
<!-- model_type 及 name。 -->
<!-- ***** -->

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

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

<!-- ***** -->
<!-- GenericView_Container 元素（用於 -->
<!-- GenericView 容器）可以包含任何數量的 -->
<!-- GenericView_Container 和 Device 元素。 -->
<!-- -->
<!-- 此元素需要有 model_type 和 name 屬性。 -->
<!-- containment_relation 不是必要屬性。 如果未-->
<!-- 指定，則會繼承上層的 containment_relation。 -->
<!-- Model_Attr 可用於多行 -->
<!-- 文字字串 SPECTRUM 屬性或清單屬性。 -->
<!-- ***** -->

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

```

```

<!-- ***** -->
<!-- Model_Attr 用於多行文字字串或清單 -->
<!-- SPECTRUM 屬性。 -->
<!-- -->
<!-- 此元素需要 attr_id 來指定 -->
<!-- SPECTRUM 屬性。 此元素可以包含 -->
<!-- SPECTRUM 文字字串屬性類型的多行文字字串，或 -->
<!-- SPECTRUM 清單屬性的多個 List_Value 元素。 -->
<!-- ***** -->

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

<!-- ***** -->
<!-- List_Value 用於 SPECTRUM 清單屬性值。 -->
<!-- -->
<!-- 每一個 List_Value 包含 PCDATA，而且做為 -->
<!-- 此清單屬性的一個執行個體值。 -->
<!-- ***** -->
<!ELEMENT List_Value ( #PCDATA ) >

<!-- ***** -->
<!-- 服務層級管理拓撲元素。 -->
<!-- ***** -->

<!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 ( SImContains ) #IMPLIED
            model_type     ( SM_ServiceMgr ) #IMPLIED
            >

<!ELEMENT SM_SLA_Mgr ( SM_SLA )*>

```

```

<!ATTLIST SM_SLA_Mgr
    name          CDATA          #IMPLIED
    containment_relation ( SImContainsSLAs ) #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 ( SImHasServiceComponent ) #IMPLIED
    model_type     ( SM_Service_Mgt ) #IMPLIED
>

<!-- Correlation 元素代表根模型 Correlation_Manager -->
<!-- (0x10469)。 只能包含 Correlation_Domain 元素，而且 -->
<!-- 沒有屬性。 所有 Correlation_Domain 會透過 CORRELATES 關係， -->
<!-- 而與 Correlation Manager 建立關係。 -->

<!-- 因為 Correlation_Manager 是唯一模型，此元素實際上 -->
<!-- 不會讓 SpectroSERVER 建立模型。 它代表 -->
<!-- 已存在的模型。 -->

<!ELEMENT Correlation ( Correlation_Domain )*>

<!-- Correlation_Domain 可以包含任何數量的 Device、Port、 -->
<!-- Model_Attr 或 GenericView_Container。 它們會 -->
<!-- 透過 CORRELATES 關係而與 Correlation_Domain 建立關係。 Correlation_ -->
<!-- Domain 也沒有屬性。 -->

<!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 |
                        排程
                        )*>

<!-- ***** -->
<!-- 代表 SM_Service 模型 -->
<!-- -->
<!-- Criticality 可以是下列其中一個值 -->
<!-- -->
<!-- 10 - 低 -->
<!-- 15 - 中低 -->
<!-- 20 - 中 -->
<!-- 25 - 中高 -->
<!-- 10 - 高 -->
<!-- -->
<!-- AttrToWatch 可以是下列其中一個值 -->
<!-- -->
<!-- Condition - 可用於大多數模型、原則 1-5 -->
<!-- Contact_Status - 通常用於裝置模型、原則 10-13 -->
<!-- Port_Status - 用於介面模型、原則 14-17 -->
<!-- LatestErrorStatus - 用於 RTM_Test 模型、原則 18-21 -->
<!-- 或 Response_Time -->
<!-- RM_Condition - SM_AttrMonitor 或 SM_Service 模型、原則 6-9 -->
<!-- 或 Service_Health -->
<!-- -->
<!-- MonitorPolicy_ID - 1-21 GlobalConfig 的 SLM_DefaultPolicies 索引 -->
<!-- -->
<!-- 1 - 狀況彙總 -->
<!-- 2 - 狀況備援 -->
<!-- 3 - 狀況高敏感度 -->
<!-- 4 - 狀況低敏感度 -->
<!-- 5 - 狀況百分比 -->
<!-- 6 - 服務健康情況備援 -->
<!-- 7 - 服務健康情況高敏感度 -->
<!-- 8 - 服務健康情況低敏感度 -->
<!-- 9 - 服務健康情況百分比 -->
<!-- 10 - 連絡狀態備援 -->
<!-- 11 - 連絡狀態高敏感度 -->
<!-- 12 - 連絡狀態低敏感度 -->
```



```
<!-- 或 Response_Time -->
<!-- RM_Condition - SM_AttrMonitor 或 SM_Service 模型、原則 6-9 -->
<!-- 或 Service_Health -->
<!--
<!-- MonitorPolicy_ID - 1-21 GlobalConfig 的 SLM_DefaultPolicies 索引 -->
<!--
<!-- 1 - 狀況彙總 -->
<!-- 2 - 狀況備援 -->
<!-- 3 - 狀況高敏感度 -->
<!-- 4 - 狀況低敏感度 -->
<!-- 5 - 狀況百分比 -->
<!-- 6 - 服務健康情況備援 -->
<!-- 7 - 服務健康情況高敏感度 -->
<!-- 8 - 服務健康情況低敏感度 -->
<!-- 9 - 服務健康情況百分比
<!-- 10 - 連絡狀態備援 -->
<!-- 11 - 連絡狀態高敏感度 -->
<!-- 12 - 連絡狀態低敏感度 -->
<!-- 13 - 連絡狀態百分比 -->
<!-- 14 - 連接埠狀態備援 -->
<!-- 15 - 連接埠狀態高敏感度 -->
<!-- 16 - 連接埠狀態低敏感度 -->
<!-- 17 - 連接埠狀態百分比 -->
<!-- 18 - 回應時間備援 -->
<!-- 19 - 回應時間高敏感度 -->
<!-- 20 - 回應時間低敏感度 -->
<!-- 21 - 回應時間低百分比 -->
<!-- -->
<!-- Special_Cause_List - 警報原因的清單或範圍， -->
<!-- 這些警報原因可用來指定包含或排除 -->
<!-- 影響「服務」或「資源監控器」 -->
<!-- 模型的服務健康情況。 只有在 -->
<!-- AttrToWatch 是 Condition 時才可使用。 -->
<!-- -->
<!-- Cause_List_Control - 指定如何使用 Special_Cause_List。 -->
<!-- 0 - 未使用 -->
<!-- 1 - 包含 -->
<!-- 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 |
                  排程
                  )*>

<!-- ***** -->
<!-- 代表 SM_Guarantee 模型 -->
<!-- -->
<!-- SLAControl 可以有下例值 -->
<!-- -->
<!-- 0 = 非使用中 -->
<!-- 1 = 使用中 -->
<!-- -->
<!-- ***** -->

<!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 |
                        排程
                        )*>

```

```
<!-- ***** -->
<!-- 代表 SM_Guarantee 模型 -->
<!--
<!-- GuaranteeControl 可以有下例值 -->
<!--
<!-- 0 = 非使用中 -->
<!-- 1 = 使用中 -->
<!--
<!-- GuranteeType 可以有下例值 -->
<!--
<!-- 0 = 可用性 -->
<!-- 1 = 效能 -->
<!-- 2 = 平均修復時間 -->
<!-- 3 = 最大中斷時間 -->
<!--
<!-- ServiceHealthType 可以有下例值 -->
<!--
<!-- 1 - 關閉 -->
<!-- 2 - 降級 -->
<!--
<!-- ***** -->

<!ATTLIST SM_Guarantee
    name CDATA #REQUIRED
    containment_relation ( SImIsMeasuredBy |
        SImSchedulesGuarantee ) #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 ( SImMonitors |
                          SImWatchesContainer ) #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      |
                        )*>

<!-- ***** -->
<!-- 代表 SM_Customer 模型 -->
<!-- -->
<!-- Criticality 可以是下列其中一個值 -->
<!-- -->
<!-- 10 - 低 -->
<!-- 15 - 中低 -->
<!-- 20 - 中 -->
<!-- 25 - 中高 -->
<!-- 10 - 高 -->
<!-- -->
<!-- ***** -->

<!ATTLIST SM_Customer
    name                CDATA #REQUIRED
    containment_relation ( SImAgreesTo |
                          SImUses ) #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
>

<!-- ***** -->
<!-- 代表 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 >

```

附錄 C：XML 範例

本節包含 XML 範例來協助您使用 DTD。

附註：每個範例中以粗體來強調元素名稱。以粗體顯示名稱是爲了讓範例易於閱讀，不表示 XML 輸入檔必須設定這種格式。

範例 1：匯入到拓撲檢視

此範例顯示基本輸入檔，該檔案將資訊匯入到 CA Spectrum 拓撲檢視。此檔案會在 [拓撲] 檢視中建立 Network 容器模型。Network 容器中建立一個 LAN 容器模型。LAN 容器內建立兩個裝置。DNS 名稱 deadlock 會識別一個裝置，而 IP 位址識別另一個裝置。

Topology 元素的 complete_topology 屬性設定爲 False。在此案例中，CA Spectrum 會尊重拓撲檢視中其他先前存在的模型。因此，此檔案只會針對 XML 檔中所列出，但尚未模型化的項目建立模型。建立的模型會依檔案中的指定而放入拓撲階層中。先前存在拓撲階層中的模型不會被重新搜索到，而會移至檔案中指定的容器。

附註：complete_topology 設定爲 false 時，已存在容器中但未列在匯入檔中的模型，不會傳送到「尋獲失物」。complete_topology 設爲 true 時，這些模型會傳送到「尋獲失物」。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- ***** -->
<!-- 這部份是用於拓撲檢視匯入 -->
<!-- ***** -->
<Topology complete_topology="false">
  <Device ip_dnsname="10.253.9.17" model_type="GnSNMPDev"
    community_string="public"/>
  <Device ip_dnsname="nmc55-5" />
  <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>
```

範例 2：建立連線

下列範例顯示在兩個 ATM 電路之間建立連線，以及在兩個框架轉送電路之間建立連線。範例也顯示 FrameRelay DLCI 連接埠與 ATM VCL 連接埠之間的連線。

```
<?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="nmc55-5">
      <Port identifier_name="frCircuitTableInstance"
        identifier_value="4.161"/>
    </Device>
  </Connection>
</Topology>
```

```

<!-- ***** -->
<!-- FrameRelay DLCI 連接埠與 -->
<!-- ATM VCL 連接埠之間的連線。 -->
<!-- ***** -->
<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>

```

範例 3：更新和終結

此範例顯示 Update 和 Destroy 元素的用法。

Update 元素包含 Location_Container 元素。此範例使用 Location_Container 元素的 name 和 model_name 屬性來更新模型名稱。name 屬性設為等於目前名稱，並識別要更新的模型。model_name 屬性將 name 屬性的值更新為 Peace2。

Update 元素也包含 Device 元素和 Port 元素。identifier_name 和 identifier_value 屬性用來識別要更新的連接埠。其他指定的屬性是需要更新值的屬性。連接埠模型名稱變更為 port 2，而 poll_status 變更為 False。

Destroy 元素刪除裝置模型 deadlock。任何與 deadlobk 相關聯的連線或連接埠會自動終結。同時也刪除 Building 容器模型 Durham。Durham building 容器內的任何模型會傳送到「尋獲失物」。

Destroy 元素也會移除裝置 nmcss52-5 上所指定連接埠與 nmcss52-3 上所指定連接埠之間的連線。

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

<Import>

<!-- ***** -->
<!-- 模型更新.....-->
<!-- ***** -->

    <Update>
<!-- ***** -->
<!-- 將容器 Peace 的模型名稱從 Peace 變更為-->
<!-- Peace2 .....-->
<!-- ***** -->
        <Location_Container model_type="Building" name="Peace"
            model_name="Peace2"/>
<!-- ***** -->
<!-- 在裝置 nmcss52-5 上更新連接埠 ifIndex=2 -->
<!-- ***** -->
            <Device ip_dnsname="nmcss52-5">
                <Port identifier_name="ifIndex" identifier_value="2"
                    model_name="port 2" poll_status="false" />
            </Device>
        </Update>
<!-- ***** -->
<!-- 終結模型和連線。 -->
<!-- ***** -->

        <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>
```

範例 4：建立、更新和終結

下列 XML 檔說明 DTD 包含的元素的大部份功能。此檔案會在拓撲和位置檢視中建立資料、建立連線、更新屬性，以及終結模型和連線。

XML 檔的第一段會在拓撲檢視中建立模型以及這些模型之間的連線。區段的開頭是 **Topology** 元素 `<Topology...>`。首先建立容器和裝置模型，接著建立連線。Topology 元素關閉時 (`</Topology>`)，此區段就結束。

Topology 元素關閉後，有另一個區段多建立了一個連線。此區段示範，不在 Topology 元素內鑲嵌 Connection 元素，也可以建立連線。

檔案的下一段開頭是 **Location** 元素 (`<Location...>`)。此區段示範在位置檢視中建立容器和裝置模型。Location 元素關閉時 (`</Location>`)，此區段就結束。

下一段的開頭是 (**Update** 元素 `<Update>`)。此區段中修改了容器、裝置及連接埠的屬性值。您無法從 XML 檔中看出所代表每一個元素的目前屬性值。因此，要分辨正在更新的元素並不容易。一般來說，每一個元素中都有個屬性來唯一識別要更新的模型或連接埠。其餘屬性是指定來更新值。例如，第一個元素是 **Location_Container** 元素。name 可屬性唯一識別模型。指定 **model_type** 屬性是要更新其值，可能是從 **Region** 更新為 **Building**。Update 元素中唯一可定義的階層是 **Device/Port** 階層，其中指定您要更新的連接埠。Update 元素關閉時 (`</Update>`)，此區段就結束。

此檔案的最後一段使用 **Destroy** 元素。Destroy 元素會刪除：

- 位於 10.253.9.19 的裝置
- 容器模型 Durham
- 裝置 nmc55-5 與位於 10.253.9.17 的裝置上的所指定連接埠之間的連線

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- ***** -->
<!-- 這部份是用於拓撲檢視匯入 -->
<!-- ***** -->
    <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="nmc55-5" />
    </Topology>
</Import>
```

```

<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="nmc55-5">
    <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>
<!-- ***** -->
<!-- 這部份是位置檢視匯入 -->
<!-- ***** -->
  <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>

```

```

    <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>
<!-- ***** -->
<!-- 這部份是用於更新模型 -->
<!-- ***** -->
  <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" />
    <!-- ***** -->
    <!-- 這部份是用於更新裝置 nmcss52-5 上 -->
    <!-- 的連接埠 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>

```

```
<!-- ***** -->
<!-- 這部分是用於刪除模型和連線 -->
<!-- ***** -->
  <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>
```


附錄 D : .modelinggatewayresource.xml

本節包含 .modelinggatewayresource.xml 檔的副本。但是，此副本可能不是此檔案的最新版本。如需最新版本，請使用模型化閘道工具組實際隨附的檔案。

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

<TopologyImportExportResourceFile>

<!-- ***** -->
<!-- 用於拓撲匯出和匯入的 -->
<!-- SPECTRUM 屬性名稱和 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_Threshold             = "0x450002c"
MTBF_Threshold            = "0x4500032"
MTTR_Threshshold         = "0x450002f"

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

/>

<!-- ***** -->
<!-- 拓撲匯入 XML 檔中可使用的 SPECTRUM -->
<!-- 模型類型名稱和控制代碼。 -->
<!-- ***** -->

<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"
    MaintenanceScheduledBy      = "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"

/>

<!-- 當 do_not_process_pre_existing_devices_under_container_node    -->
<!-- 設為 true 時，如果發現容器元素下的裝置    -->
<!-- 已存在 Spectrum 中，則模型化閘道不會處理該    -->
<!-- 裝置，例如更新屬性、建立連線等等。    -->

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

/>

<!-- ***** -->
<!-- -->
<!-- 這是模型化閘道匯出配置。 -->
<!-- -->
<!-- ExportConfiguration 是控制要匯出什麼項目 -->
<!-- 的配置。 -->
<!-- -->
<!-- export_devices：是否匯出裝置模型 -->
<!-- -->
<!-- export_containers：是否匯出容器模型 -->

```

```

<!--                                     -->
<!-- export_port_attributes：是否匯出連接埠屬性    -->
<!--                                     -->
<!-- export_links：是否匯出裝置連結                -->
<!--                                     -->
<!-- export_topology_layout：是否匯出裝置和容器    -->
<!--                                     x,y 座標      -->
<!--                                     -->
<!-- export_annotation：是否匯出注釋              -->
<!--                                     -->
<!-- export_WA_Link_models：是否匯出 WA_Link 模型。  -->
<!--                                     如果不匯出，則將 WA_Link -->
<!--                                     模型視為透明。 兩個裝置 -->
<!--                                     之間透過 WA_Link -->
<!--                                     所建立的連結會匯出為 -->
<!--                                     直接連結。 -->
<!--                                     -->
<!-- export_spectrum_settings：是否匯出 SPECTRUM 設定， -->
<!--                                     例如錯誤隔離、 -->
<!--                                     自動搜索、 -->
<!--                                     VNM 控制等... -->
<!--                                     -->
<!-- export_user_models：匯出 SPECTRUM 使用者模型化、 -->
<!--                                     使用者授權、權限等等。 -->
<!--                                     -->
<!-- export_service_modeling：匯出 SPECTRUM 服務模型化 -->
<!--                                     -->
<!-- export_schedules：匯出 SPECTRUM 排程。 -->
<!--                                     -->
<!-- export_discovery_configs：匯出自動搜索的 -->
<!--                                     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 中要匯出的根容器。    -->
<!--                                     -->
<!-- ***** -->

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

<!-- ***** -->
<!--                                     -->
<!-- DeviceExportAttributes 是要匯出的裝置屬性 -->
<!-- 清單。 如果上面未指定屬性 ID， -->
<!-- 則必須指派 "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 是要匯出的容器 -->
<!-- 屬性。 如果上面未指定屬性 ID， -->
<!-- 則必須指派 "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 是要匯出的連接埠 -->
<!-- 屬性。 如果上面未指定屬性 ID， -->
<!-- 則必須指派 "attribute_id"。 -->
<!-- -->
<!-- 當 export_changed_attribute_only = "true" 時， -->
<!-- 只會匯出不等於預設值的連接埠 -->
<!-- 屬性。 否則，所有指定的連接埠 -->
<!-- 屬性都會匯出。 -->
<!-- -->
<!-- ***** -->
```

```
<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" />
</SpectrumConfigurationExport>
```

```
<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>
```