DITA — 您不可不知的四個字母

簡介

DITA 是近來 XML 發行技術中屬一屬二的創新發明。若您正在使用或計畫使用 XML 發行技術文件，您遲早會接觸到 DITA。

DITA 是 “Darwin Information Typing Architecture” (達爾文資訊分類架構) 的縮寫，IBM 在美國先進結構化資訊標準組織 OASIS (http://www.oasis-open.org) 的贊助下開發了此一技術，造福 XML 使用社群 (OASIS 設有 DITA 技術委員會，詳情請見：http://xml.coverpages.org/DITA-OASIS-CFP.html)。

本文件並不是技術導入人員的「使用指南」(How To)，亦不含任何高階技術資訊。IBM 提供了豐富、精闢的 DITA 技術資訊，網址為：http://www-106.ibm.com/developerworks/xml/library/x-dita1/。

文中雖提及諸多與 DITA 相關的 IBM 人員和資源，然而，本文作者必須對錯誤、疏漏或誤釋負完全責任。

目錄

簡介 1
DITA 的重要性 3
切合需要的發明 — DITA 的起源 3
進入 DITA 的世界 4
DITA 規格化結構 — 滿足您的需要 5
重複使用 — 真的還假的？ 5
設計理想的資料模型 5
規格化結構的作用 6
更容易編製和理解 7
Topic 的組成項目 7
如何分類資訊 7
提高資訊品質 8
DITA 適用於軟體、UI 和格式設定 9
掌控您的領域 9
積沙成塔 9
軟體元素 10
UI 和程式設計元素 10
讓 DITA 順利運作 10
完成 DITA 內容建立！下一步？ 10
完全整合 — PTC 的 DITA 支援功能 11
結語 12

DITA 的重要性

為了處理 XML，DITA 內建了幾個涵義深遠的簡單結構。定義 DITA 相對簡單，描述其涵義則較困難。因此，我們從定義開始說明。

DITA 是以 XML 為基礎的技術資訊發行架構。它某些部分的功能類似 DocBook — 這個工具於多年前推出，也用於發行 XML 技術手冊，特別是電腦軟硬體文件 (在進行這份白皮書的前置研究工作時，我們瀏覽了 http://www.namahn.com/resources/notes.htm 這個網頁上許多有趣的資源，並發現幾篇值得一讀的白皮書，其中有一篇與 DITA 相關的白皮書，更具技術性，也是更早期的文件，但它針對 DITA 和 DocBook 做了不錯的比較)。

DITA 有兩個主要特色：

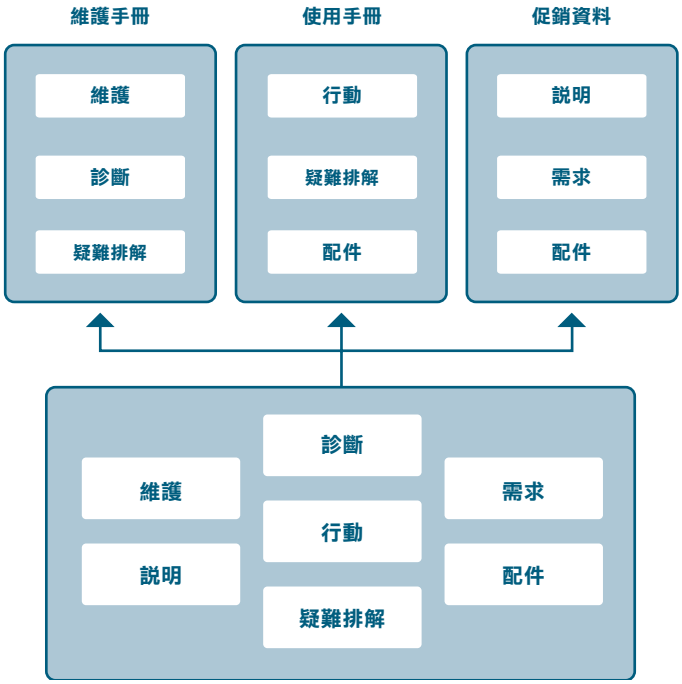
- 模組化 — DITA 定義了 Topic (主題) DTD，支援以模組化的方式建立資訊。一個主題代表一個資訊元件，並非完整的文件；而且一個主題涵蓋一個特定的議題範圍 (例如，這份白皮書可以分割為數個主題：簡介、概觀、來源等)。

  DITA 也定義了將主題結合為文件的機制。這個機制稱為「對映檔」(Map)，您也可以用它來定義主題階層。舉例來說，一本書的對映檔可能產生以下階層：章、節和段。

- 彈性調整 — Topic DTD 類似 HTML，會指定一組通用元素 (Element)，且每個元素的格式設定都不同，例如：標題、段落和清單。為了按照您的需求調整 Topic DTD，DITA 定義了「規格化結構」(Specialization) 機制，可以讓您定義新的標籤；這些新標籤的行為和屬性 (Property) 將繼承 Topic DTD 標籤。

  採用規格化結構後，具備 DITA 感知功能的下游應用程式便能依屬性判斷未知的標籤，再將其視為繼承標籤處理舉例來說，您可以建立一個標籤，將其命名為 Procedure (程序) 並繼承 Ordered List (排序清單)；然後建立另一個標籤，將其命名為 Step (步驟) 並繼承 List Item (清單項目)。您雖然想在應用程式中為 Procedure 和 Step 加入特殊的處理，不過，DITA 感知應用程式卻完全無法辨識這些標籤，於是將它們視為 Ordered List 和 List Item 處理。比方說，DITA 感知發行應用程式無法辨識 Step，因此會依據 List Item 設定其格式。

即使您瞭解 DITA 的定義，您仍不具備瞭解其涵義的背景知識。因此，我們現在開始說明。



採用 DITA 時，您可以建立可重複使用的資訊模組，再將它們彙編為不同用途的文件。

切合需要的發明 — DITA 的起源

我們現在重新由 XML 發行系統的主要目標開始說明：

- 重複使用 — 為了消除重複、提高正確性以及降低更新資訊的成本，XML 將協助您重複使用及再利用資訊來建立單一資訊來源，如此一來，內含變更資訊的文件就會依據來源變更自動更新。
- 共用 — 使用 XML 建構資訊時，組織內外的其他群組可以將您的資訊自動整合到他們的處理中，提高資訊的價值。
- 關聯性 — XML 有助於您建立資訊模組，之後，您便能依據個別需求自動彙編這些模組，進而篩選並提供必要的資訊。

- 自動化 — 若要以符合成本效益的方式達成這些目標，自動化可說是成功的關鍵。XML 允許您強制執行自動化流程所需的絕對限制結構，進而達成自動化的目標。

  自動化所需的絕對限制結構由「資料模型」定義 — 亦即 DTD 或 Schema，將指定文件內允許的標籤以及標籤的使用方式。

現在，請想像您是某家大公司的員工，該公司的產品線種類繁多，而您負責監督每一種產品的技術文件發行。XML 不但在內容編製、翻譯和發行效率驚人，還能讓貴組織提供更為正確、即時和相關的資訊給客戶，因此獲得你的青睞。

您的系統核心為資料模型，專為公司內的群組發行技術文件而設計。採用全公司適用的發行方法，不但可以降低取得發行工具及研發發行應用程式的成本，還可以統一公司形象，並針對客戶的個別需求，將多種產品結合為自訂文件。

然而，隨著新群組的設立，資料模型也跟著漏洞百出。這些新群組的需求超出您的預期，且紛紛要求您變更系統。每項變更不只會影響資料模型，還會影響所有依賴資料模型的下游應用程式 — 其中又以彙編和發行最為顯著。每項變更可能影響許多發行樣式表及其他處理邏輯，因此需要更新、測試、記錄和重新部署應用程式。

隨著越來越多的群組上線及要求變更，加上現有群組發現新的機會而要求變更，您終於瞭解變更要求將無止無休，且您的進度落後而永遠無法趕上。

進入 DITA 的世界

基於這些挑戰，您決定採用新的方法。您需要尋找一種足以處理數種相衝突需求的方法：

- 滿足組織內眾多領域的需求
- 能夠依據業務需求的變更迅速、輕鬆地調整
- 支援高度模組化的資訊，加強提升重複使用率和個人化的程度
- 簡化內容和程序，協助內容編製者迅速提高生產力

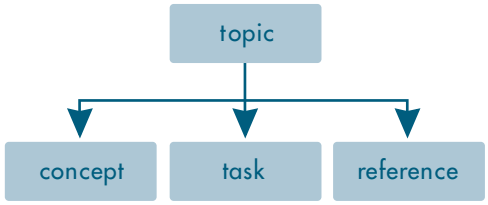
一開始，您建立了發行技術文件所需的 DTD，其中納入所有的通用格式設定結構。此時，HTML 當然是最好的起點，因為它的彈性特點已由數百萬網頁證實。標題、段落、項目符號清單、斜體字等等 — HTML 全都能表示。

您的新 DTD 不含任何企業專用標籤 — 只有類似 HTML 的「通用」標籤。這些通用標籤控制資訊的外觀，但幾乎完全無法表示資訊的「語意」(Semantic)，也就是資訊的意義。此外，新 DTD 也不含文件階層，只代表一種資訊模組 (稍後，您將使用一種將模組結合成完整文件的方法，並將文件傳遞給客戶)。

您將 DTD 命名為 Topic，接著再設計一套機制，以便您依據特定需求輕鬆地調整 Topic。受 1990 年代「架構格式」(Architectural Form) 的啟發，您建立了定義新標籤的方法，而從現有的標籤繼承屬性。其他應用程式若能瞭解您的語法 (依據現有標籤定義新標籤)，它就能夠處理您的新標籤。

由於您正在建立技術文件，因此您轉而注意這類文件所需的特定資訊類型。以資訊架構的原則為基礎，輔以與有效傳達資訊給使用對象的大量研究、理解和體驗，您定義了三個 Topic 的規格化結構，做為文件的基本元件：

- Concept
- Task
- Reference



Concept (概念)、Task (工作) 和 Reference (參照) 是 Topic 的規格化結構，分別採用不同的方式提供某一主題的相關資訊。雖然多數 DITA 感知系統都應該個別瞭解如何處理 Concept、Task 和 Reference，不過，任何 DITA 感知系統都可以處理任何的規格化結構。

這些規格化結構不僅是您的最佳創意，它們還代表了 DITA 的另一項重要創新：使用資料模型引導內容編製者建立易於編製及瞭解的資訊。

DITA 規格化結構 — 滿足您的需要

「規格化結構」是 DITA 中的一種機制，您可依需求加以自訂，同時維持現有內容和發行系統的互通性和重複使用性。此外，「規格化結構」還能在企業需求變更時降低發行應用程式的維護成本，因為在多數情況下，您可以修改資料模型而不需變更下游應用程式。

本節將說明規格化結構的影響和重要性，但為完整評估規格化結構，我們需要先瞭解重複使用資訊的基本概念，因為這是 XML 發行系統的基礎之一。

重複使用 — 真的還假的？

XML 發行技術具有許多潛在優勢，能夠以多種不同方式重複使用資訊，就是它最重要的優點。若您使用 XML，則我們將繼續說明；您可以將相同的資訊插入不同的文件，完全不需修改，然後使用不同的媒體類型發行，同時又能維持單一資訊來源，如此一來，當資訊一有變更，顯示變更資訊的所有文件便會自動更新。

這並不是神話 — 許多組織都已經達成這個重複使用資訊的目標。當您嘗試在不同類型的文件中重複使用資訊時，就會開始發生問題，尤其是資訊來源分散組織內外不同位置時。

問題在於：各部門均依本身需要建立不同的資料模型 (DTD 或 Schema)，而且其中包含特定用途的標籤。舉例來說，維修資訊可能包含 <procedure> 或 <step> 標籤，而零件目錄則包含 <partnumber>、<description> 和 <price> 標籤。

若您要在零件目錄中加入維修資訊，或在維修手冊中加入零件資訊，則必須配合新標籤變更資料模型。不只如此，您還必須變更處理資訊的所有應用程式。比方說，您必須配合新標籤變更樣式表，如此才能發行含新標籤的資訊。

若您嘗試在不同的文件類型、更多的部門和企業中重複使用資訊，這個互通性的問題會變得更加嚴重。在這種情況下，想要重複使用資訊是不可能的任務，因為調整不同資料模型的成本將超過整合不同資料模型的利益。

倘若每個部門都限制自己使用簡單、通用的標籤，例如 HTML，資訊共用應該還算簡單。不過，使用簡易資料模型卻不利於自動化和內容處理。

真讓人進退兩難！資料模型因組織需求所調整的幅度越大，資訊的價值就越高。然而，資料模型的特殊化程度越高，重複使用及共用資訊的困難度也越高。

即便存在這樣的困境，許多組織仍因工作群組或部門內的內容重複使用及自動化獲益良多。假設各個部門、甚至不同企業都可以運用這些功能，實現的利益會有多大啊！

設計理想的資料模型

理想的資料模型應該是適合全世界使用又容易學習。這是什麼樣的資料模型呢？

為了滿足所有人的需要，資料模型應該包含數十萬個標籤以及數萬種標籤組合。不過，為了大家方便學習，標籤數量必須限制在 100 個以內！

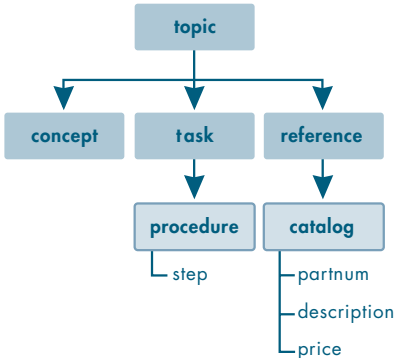
所以這是行不通的。

每個部門對本身建立的內容都有獨特的需求。不過，該資訊應該可以在組織內重複使用。如何讓建立的資料模型支援整個組織的需要，同時又方便內容編製者使用呢？

DITA 的設計師找到了巧妙的替代方案：與其建立一個試圖滿足個人需要的資料模型，不如建立一個能夠依據個人需要調整的資料模型。這就是「規格化結構」背後的設計靈感；您可以依需要使用 DITA 的「規格化結構」修改 DITA 資料模型，並採用其他的 DITA 感知應用程式整合您的內容，而且這些應用程式不需具備辨識標籤的能力。

使用 DITA 時，請從 DITA 指定的基本標籤開始。這些標籤類似 HTML，經過數百萬網頁的使用證明和好幾代的演進改良，不但具備極佳的彈性，而且包含大部分常用的格式設定與結構化標籤。

若您需要修改 DITA，請自行加入標籤。如果您在其他資料模型中加入自己的標籤，內容互通性將立即消失或降低。不過，DITA 有一個很棒的使用祕訣：當您加入標籤時，您不只指定標籤名稱，還要指定用途最接近的基本標籤。換句話說，您建立的新標籤將繼承現有標籤的屬性和行為 (這就是 DITA 中 D 的由來：代表生物學中的達爾文遺傳理論)。



如前例所示，您可以透過既有的規格化結構來建立新的規格化結構：procedure 是 task 的規格化結構，catalog 則是 reference 的規格化結構。在每個規格化結構中，您還可以建立新標籤來滿足特定的需求，例如每個標籤都是 topic 下之標籤的規格化結構或 topic 的規格化結構。

舉例來說，假設您要建立 <partnumber> 標籤。由於零件編號可能是內置標籤，因此，您可以設定其繼承基本 DITA 資料模型中的 <emph> (代表「強調」) 標籤。您仍可設定應用程式以特殊的方式處理 <partnumber> 標籤。舉例來說，您可以設定不同的格式設定，並依據資料庫自訂驗證。

採用這種方式的優點，在於即使別人完全不瞭解 <partnumber> 標籤，仍可整合及重複使用您的內容，而不需變更基礎應用程式。該怎麼做呢？您的資料模型中包含繼承資訊，因此即使其他 DITA 感知應用程式無法辨識 <partnumber>，仍會將其視為 <emph> 處理。

再舉個例子：假設您要設定一個包含連續步驟的程序。您可以設定 <procedure> 標籤繼承 <ol> (代表 ordered list，屬於編號清單) 標籤，並設定 <step> 標籤繼承 <li> (代表 list item) 標籤。

規格化結構的作用

規格化結構符合 XML 和 XSL 標準 — DITA 文件與這些標準完全相容。

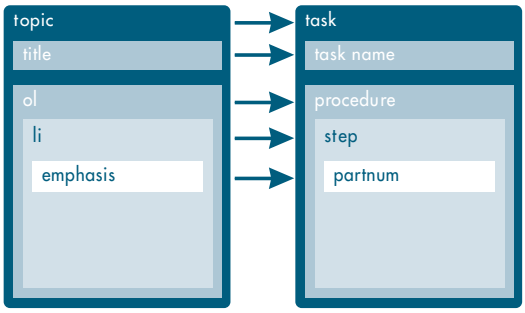
若要依據您的需要調整 DITA，請由基本 DITA 資料模型中的 Topic 開始，這是所有規格化結構的起源，並指定適當的名稱。舉例來說，假設您要建立名叫 Task 的資料模型。建立 Task 資料模型時，請由 Topic 中的標籤開始，然後再加入您自己的標籤。接著，假設您加入 <procedure> 和 <step> 到您的資料模型。

當您依循 DITA 模型定義新標籤時，您需要指定新標籤屬性所繼承的標籤。這項資訊會儲存在 DITA 設計師稱為 class 的特殊屬性 (Attribute) 中。class 屬性中包含繼承資訊。在 DTD 中，class 屬性定義語法使用 ATTLIST 關鍵字；以下範例將定義 <step> 標籤的繼承資訊：

<!ATTLIST step class CDATA “- topic/li task/step”>

此 ATTLIST 宣告不僅宣告了 <step> 標籤的 class 屬性，還指定了屬性的預設值。為了和 DITA 方法相容，您禁止內容編製者在編輯時變更此屬性。

在前述範例中，<step> 的 class 屬性設定為 - topic/li task/step。對 DITA 感知應用程式來說，- topic/li task/step 表示如果應用程式可以辨識 Task 資料模型，則應將 <step> 視為 <step> 處理。不過，如果應用程式無法辨識 Task，則應將 <step> 視為 Topic 資料模型中的 <li> 處理。



DITA 基本元件 Task 規格化結構

規格化結構是以 DITA 中的基本元件做為基礎。如果應用程式或樣式表無法辨識規格化結構中的標籤，就會將其視為 DITA 來源標籤處理。

為了發行以 DITA 為基礎的內容，您依據 class 屬性設定樣式表，而非標籤。基本資料模型 (Topic) 的樣式表只要檢查 class 屬性，就可以處理任何的規格化標籤。

更容易編製和理解

我們現在來仔細看看 IBM 所定義的三個規格化結構 — concept、task 和 reference，並瞭解它們對內容編製和理解的影響。

Topic 的組成項目

DITA 的基本元件是 Topic，代表特定目的的簡短描述。

Topic 由一組用來組織和呈現資訊的基本標籤組成；Topic 中的部分標籤包括：

組織標籤	
標籤名稱	**說明**
<title>	標頭或標記；可以是主標題 (在 <topic> 下) 或副標題 (例如在 <section> 下)
<shortdesc>	連結預覽和搜尋的簡短說明
<section>	topic 的一部分；topic 可以包含多個 section (小節)，但 section 不得包含另一個 section
<body>	包含 topic 的主要內容
<p>	段落
	未排序清單 (項目符號清單)
	排序清單 (編號清單)
<table>	表格

呈現標籤	
標籤名稱	**說明**
<b>	粗體
<i>	斜體
<u>	底線
<tt>	電傳打字 (等寬字形)
	未排序清單 (項目符號清單)

Topic 中不含特定產業、應用程式或文件類型的專用標籤。由於 DITA 您允許依據特定需求調整 Topic，因此，您可以在 Topic 下自行加入規格化結構。

加入新標籤時，您還必須指定與新標籤類似的現有標籤 (請參閱下表中的「繼承來源」欄)。如此一來，即使其他 DITA 感知應用程式無法辨識新標籤，也可以處理您的內容 — 它們會採取遞補的方式，將新標籤視為繼承來源標籤處理 (如下表所示，DITA 感知應用程式無法辨識 <prereq> 時，會將其視為 <section> 處理)。

如何分類您的資訊

當您建立 Topic 的規格化結構時，您正在建立新的「資訊類型」(又稱為 infotype)。舉例來說，IBM 建立的 Topic 規格化結構稱為 Task，其中包含完成程序所需執行的步驟。換句話說，Task 是一種「資訊類型」，其中包含的規格化標籤有：

Task 標籤		
標籤名稱	**說明**	**繼承來源**
<prereq>	必要條件：使用者開始工作前需要知道或執行的事項	<section>
<steps>	使用者完成工作所需採行的連續步驟	
<result>	執行工作的預期結果	<section>

DITA 允許您調整 Topic 或其他「資訊類型」來自行建立「資訊類型」。比方說，假設您要建立新的「資訊類型」：Repair Procedure (修復程序)。您可以依需求從 Topic 開始調整，不過，從 Task 開始可能比較容易，因為它已經包含您需要的部分標籤。

建立文件時，您需要彙編一或多個 topic 或其他「資訊類型」。文件可以包含任何「資訊類型」的組合。舉例來說，一份文件中可以包含數個 Topic、Task 以及您自己的「資訊類型」。

提高資訊品質

以資訊架構的原則為基礎，輔以大量的相關研究、理解和體驗，IBM 特別針對技術文件定義了三種「資訊類型」— Concept、Task、Reference。

技術文件還存在其他資訊架構。例如 Information Mapping 這家公司便開發了一套方法並發行了大量的研究文獻，顯示撰寫技術文件 (以字數和完成時間衡量) 與閱讀技術文件 (以時間和理解度衡量) 兩方面都有長足的進步。如需 Information Mapping 的詳細資訊，請參閱：http://www.informationmapping.com/。如需檢視該公司有關技術文件改良的研究文獻，請造訪：http://www.informationmapping.com/documents/IMI%20history%20and%20results.pdf (這個連結將開啟 Adobe Acrobat® 文件)。

無論方法為何，它們都有一些相同的特點：

- 指定內容編製者可以建立的資訊類型 (例如：concept、task、reference)
- 建立資訊模組，將資訊劃分為多個可重複使用的小元件
- 讓內容編製者一次專心建立一個元件
- 讓發行自動化，為內容編製者免除設計和格式設定的負擔
- 將資訊分類並統一呈現方式，方便讀取程式快速判斷資訊
- 便於多份文件中重複使用資訊元件，且便於跨組織共用

DITA 的概念非常簡單，技術文件只有三種基本資訊類型。這些類型都可以針對特定的產業和文件再進行特殊調整，不過，一本書中最後只能包含三種資訊類型。

以下為各種 DITA 資訊類型的說明：

- Concept：Task 或 Reference 的概觀。Concept 通常十分簡短，部分原因在於內容編製者只能在 section 或 example (範例) 後加入其他 section 或 example。
- Task：包含讀取程式執行程序所需的步驟。Task 中有許多規格化標籤，可以支援各種指示呈現方式以及背後的資訊。
- Reference：描述一組事物的特性，例如產品清單、定義或程式指令。

這些資訊類型各有不同的特性，但共用許多相同的標籤。除了先前列示的組織和呈現標籤，還有連結 (交叉參照和相關連結)、搜尋、瀏覽、關鍵字等其他標籤。

DITA 非常有趣，它還可以引導您計算文件中的某些統計數據，例如：

- 元件中的平均字數；這個數據有助於辨識字數太少或太多的元件 (您可能會發現，不同資訊類型的目標大小也應該不同)
- 書籍中出現的資訊類型比例；這個數據有助於辨識書本中是否強調的參考資料太多或是強調的概念太少

簡單來說，在技術文件中套用 DITA 方法時，您的手冊已經有初步的結構。您可以直接依需要調整 DITA 的三個「資訊類型」— concept、task 和 reference。內容編製者在撰寫 XML 文件時，將能更輕鬆地進行標籤的管理工作；此外，由於他們不再需要設定格式，而且能夠使用定義完整、經實際驗證的範本，所以生產力也將隨之提高。最重要的是，讀取程式將能更迅速地找到需要的資訊，且需要讀取的字數和自定組織結構同時減少。

DITA 適用於軟體、UI 和格式設定

「領域元素」(Domain Element) 是 IBM 針對特定主題領域所定義的規格化標籤。這些標籤不但是針對各種主題 (包括程式設計、軟體、使用者介面和格式設定) 進行特殊調整的最佳起點，也可以示範如何將 DITA 的規格化結構支援套用到您需要使用的任何應用程式。

掌控您的領域

DITA 為採用主題導向方法建立資訊的最佳起點，不過，若要在您的領域中成功套用 DITA，關鍵在於加入能滿足特定需求的「規格化結構」。

舉例來說，如果您要發行服務手冊，您可能需要建立包含已排序工作清單的 procedure 標籤。套用 DITA 時，您需要在現有標籤下建立規格化結構 <procedure>。由於程序看似編號清單，因此 <procedure> 屬於外觀類似編號清單之 <ol> (排序清單) 的規格化結構。

<procedure> 可用於多種服務手冊，不過，您可能還需要建立專門適用於您的產業或組織的特定標籤。舉例來說，如果您經營的是軟體公司，您可能必須將應用程式發展介面 (API) 的相關資訊記錄下來，並針對這樣的需求建立一些規格化標籤。

這類的規格化標籤屬於某個「領域」，也就是主題領域的另一種花俏說法。設計 DITA 時，IBM 還定義了涵蓋特定主題領域的多組規格化標籤「領域元素」。這些主題領域包括程式設計、軟體、使用者介面和格式設定。下節將說明各組「領域元素」的用途。

從現有的「領域元素」開始有助於縮短自行開發應用程式的時間，並能提高同領域共用內容的機會。當別人都使用 <parml> 時，您還需要建立意義相同的 <parameterlist> 標籤嗎？

積沙成塔

我們現在從設定格式的「領域元素」開始說明。這些標籤稱為「印刷元素」(Typographic Element)，可以讓內容編製者控制文字的格式。

等一下！如果我們建立純粹用於控制元素外觀的標籤，那不是違反了 XML 區隔內容和格式設定的原則嗎？沒錯。倘若您支持區隔原則，則您可以在「印刷元素」下建立其他的規格化結構來擴充標籤的意義。舉例來說，您可以在指定斜體文字的 <i> 標籤下建立 <booktitle>、<emph>、<foreignword> 等規格化標籤，直到您納入斜體文字的各種情況。至於值的數目是否足以區隔不同意義的斜體文字，則須視您的需要而定 (而非 XML 的「單純原則」)。

這些新標籤可以稱為「語意」標籤，因為它們指定文字的意義，而非文字的外觀，如此一來，您稍後就能靈活調整格式設定，不需重新指定標籤。簡直是一種 XML 病毒！

不過，某些情況根本不值得建立規格化標籤，此時，使用「印刷元素」是最適合擷取文字意義的遞補方式。如果您決定直接開放內容編製者使用「印刷元素」，我們也許不會責備您，不過，我們也許會，請仔細考慮。

所以我們現在來看看整個清單：

印刷元素	
標籤名稱	**說明**
<b>	粗體
<i>	斜體
<u>	底線
<tt>	電傳打字 — 選取 Courier 之類的等寬字型，讓字元間隔相同；適用於需要垂直對齊字母時
<sup>	上標 — 縮小文字並讓其位置高於周圍的文字；適用於註腳以及數學式，例如 22 = 4
<sub>	下標 — 縮小文字並讓其位置低於周圍的文字；適用於化學式，例如 H2O

軟體元素

若您嘗試描述軟體程式的運作方式，以下「領域元素」將非常實用：

軟體元素	
標籤名稱	**說明**
<msgph>	訊息文字 — 包含程式發送的訊息
<msgblock>	訊息區 — 包含多行訊息或多則訊息
<msgnum>	訊息編號 — 如果有的話
<cmdname>	指令名稱
<varname>	變數 — 包含輸入程式的變數
<filepath>	檔案路徑 — 包含檔案名稱和位置 (選擇性)
<userinput>	使用者輸入 — 包含使用者輸入程式的內容
<systemoutput>	系統輸出 — 包含任何類型的程式輸出；(<msgph> 的涵蓋範圍較 <systemoutput> 小)

使用這些元素必須遵守若干規則。舉例來說，根據這些「領域元素」的定義，msgph、msgblock、filepath、userinput 和 systemoutput 可含 varname，filepath 卻不可以。

這些結構規則幾乎和標籤名稱一樣重要，甚至更為重要。轉換標籤及還原相對簡單，轉換結構及還原則較困難。

UI 和程式設計元素

除了「領域元素」之外，還存在著所謂的「使用者介面」元素和「程式設計」元素。

您可以使用「使用者介面」元素詳細描述電腦程式中的使用者輸入。這些元素包括：功能表、快速鍵、電腦螢幕和使用者介面控制項 (如：按鈕和功能表項目)。

「程式設計」領域與其他領域不同，內含 25 個元素，數目較多，因為程式語言中有極多記錄程式碼撰寫方式的不同部分。有了關鍵字、程式碼、變數和參數等元素，您可以描述程式語言中的所有細節。

讓 DITA 順利運作

根據設計，DITA 內容由許多資訊區塊組成，而非書籍。應該如何將這些區塊轉換成連貫的資訊呢？例如：線上說明檔或書面文件。這個時候，您需要使用對映檔。

最後，又應於何時使用支援工具建置 DITA 應用程式、建立 DITA 內容、將內容彙編為連貫的資訊以及將內容發行到不同的媒體呢？這個時候，PTC 正好派上用場。

完成 DITA 內容建立！下一步？

DITA 推廣的主題導向內容有一個最大的優點，那就是內容編製者需要撰寫的區塊相對縮小。假設內容編製者原本必須撰寫 1000 頁，現在只需要處理短短幾頁的資訊元件。

若要將這些區域彙編為文件，您可以使用 DITA 定義的「對映檔」指定要納入的區塊、區塊的順序以及區域的階層關係。

舉例來說，假設您撰寫了下列區塊：

- Topic：化油器
- Concept：化油器的運作方式
- Task：調整您的化油器
- Task：替換您的化油器
- Reference：化油器和其他引擎元件的關係

為了將這些區塊彙編到文件中，您需要將它們列示於對映檔中。您不但可以在對映檔中列示要納入的區塊，還可以任意指定階層關係。

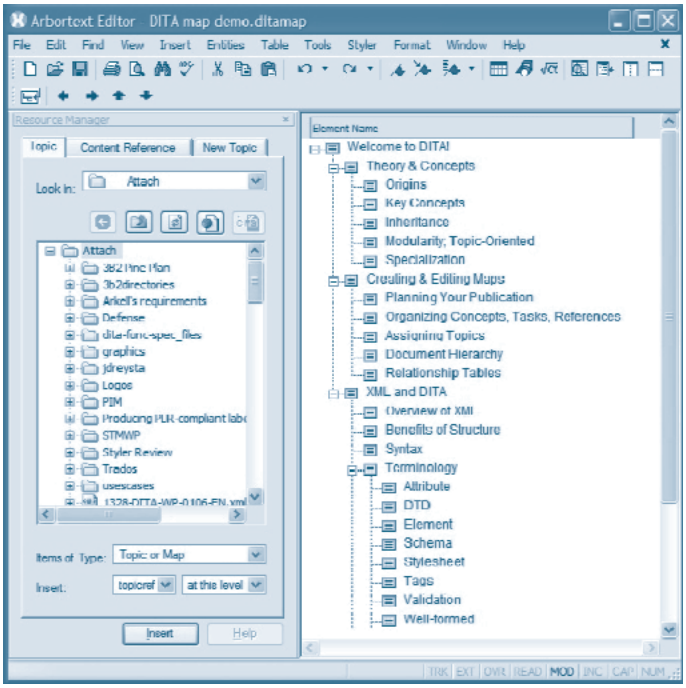
舉例來說，本例中的最後四個區塊從屬於第一個區塊，如以下階層所示：

- Topic：化油器
  - Concept：化油器的運作方式
  - Task：調整您的化油器
  - Task：替換您的化油器
  - Reference：化油器和其他引擎元件的關係

對映檔中的主要元素是 <topicref>，這個 Inclusion 機制 (類似檔案實體或 XInclude) 可以指定目前文件中要納入的區塊。

以下影像顯示化油器文件的對映檔。若要建立這個階層，您可以在 <topicref> 下加入 <topicref> 來建立巢狀結構，表示第二個 <topicref> 是第一個 <topicref> 的從屬項目。

以下範例中，「化油器」的 <topicref> 下包含所有剩餘區塊的 <topicref> 元素，這些下層區塊都是上層區塊的從屬區塊：



Arbortext 5.3 版隨附的 Arbortext Editor 提供數種工具，可以讓您更輕鬆地建立 topic (或 topic 的規格化結構，如 Concept、Task 和 Reference)、列示資訊集 (如書籍或線上說明檔) topic 的「對映檔」，並簡化瀏覽 topic 的方式。

全面整合 — PTC 提供的 DITA 支援功能

雖然 DITA 的原始意圖為遵守現有的標準，不過 DITA 仍引進了部分功能，必須有其他的處理支援。PTC 預測 DITA 將成為多種應用程式的重要架構，因此決定進行大規模投資來加入 DITA 特殊功能。

PTC 在 Arbortext 系列產品中加入並改良數種 DITA 支援功能：

- 編製：我們在 Arbortext Editor 中加入數種建立和編輯 Topic 與 Map 的功能，為您提供了一個易學易用的高效率環境。這些功能包括：以拖放方式在 Topic 中加入元件或在 Map 中加入 Topic；在 Map 中執行拼字檢查以及尋找/取代所有的 Topic；使用新的 Resource Manager 放置可重複使用的元件；以及使用新的 Column View 以檢視目錄的方式來檢視對映檔。
- 彙編及發行：Arbortext Publishing Engine 採用您的 DITA 對映檔來彙編文件，然後將這些文件發行到各種平面與電子媒體上。
- 規格化結構：Arbortext 系列產品支援 DITA 規格化結構功能，因此，您可以使用單一樣式表來控制 Topic、Task、Concept、Reference 和其他規格化結構的樣式。雖然 XSL 樣式表已有支援規格化結構的機制，不過，我們仍必須新增內部樣式表格式的規格化結構支援，以便於編輯。
- 語意表：DITA DTD 中的部分元素無法敘述而必須以表格顯示，因此 Arbortext 產品允許您選取能以表格顯示和列印的標籤。

  舉例來說，目錄中的 <partnum>、<description> 和 <price> 標籤都必須以表格顯示。如果無法支援語意表，這些標籤將顯示為敘事內容：

  partnum **123-456** partnum

  descripition **化油器** descripition

  price **$45.99** price

  有時候資訊表格 (如下所示) 較清單或文句更為清楚。

為便於編輯，您可以設定 Arbortext Editor 以下列表格儲存格顯示標籤：

零件編號	說明	價格
123-456	化油器	$45.99

- Conref 支援：DITA 使用 conref 在文件中加入資訊元件。這是檔案實體機制和 XInclude 的替代支援，必須使用 PTC 加入 Arbortext 系列產品的特殊功能。
- Topicref 支援：DITA 使用 topicref 在對映檔中加入 Topic (或 Topic 的規格化結構，如 Concept、Task 或 Reference)。這是檔案實體機制和 XInclude 的替代支援，必須使用 PTC 加入 Arbortext 系列產品的特殊功能。
- Doctype：Arbortext 軟體中包含 Topic、Concept、Task 和 Reference 適用的一組 Doctype。Doctype 中包含一分樣式表 DTD 和該 Doctype 適用的其他自訂功能。

結語

跨企業重複使用及共用資訊是 XML 發行系統的主要優點之一。然而在多數組織中，每個部門的記錄規則和需求通常不同。若要建立便於內容編製者使用又能支援各部門需求的資料模型，往往相當累人。而且隨著內容編製者和企業需要的不斷演變，管理資料模型幾乎成為不可能的任務。

DITA 找出解決互通性問題的方法，運用獨特的資料模型滿足每個組織部門的個別需要。各資料模型的大小設有限制，非常方便內容編製者使用。不過，由於所有的資料模型都以單一全域資料模型為基礎，內容編製者仍能跨組織共用及重複使用資訊。這是因為樣式表和應用程式遇到獨特資料模型中無法辨識的標籤時，可以根據相似的全域標籤加以處理。

內容編製者的生產力將因此提高，因為他們可以重複使用任何組織部門的現有資訊，而且來源更新時，所有被重複使用到的相關內容也會自動更新。如此一來，內容編製者重建現有資訊及更新重複使用資訊的時間同時減少。重複使用加速、簡化了全域更新，可以提高資訊的一致性和即時性。最後，確認 XML 樣式表和應用程式依據全域 DITA 標籤處理未知的衍生標籤，可以確保發行自動化不因錯誤和更新而減緩了處理速度。

總之，DITA 可以協助組織實現 XML 的潛在優勢、提高內容編製者的生產力和資訊的品質，並讓發行處理的速度加快、成本降低。

© 2006 Parametric Technology Corporation (PTC) 版權所有，盜版必究。此處所描述的資訊僅作為參考之用，如有變更恕不通知，且不得將其視為 PTC 所做之擔保、承諾、條件或提議。PTC、PTC 標誌、The Product Development Company、Arbortext、Arbortext 標誌、Pro/ENGINEER、Wildfire、Windchill 以及所有的 PTC 產品名稱和標誌都是 PTC 和/或其子公司在美國及其他國家的商標或註冊商標。所有其他產品或公司名稱則為其各自擁有者的財產。