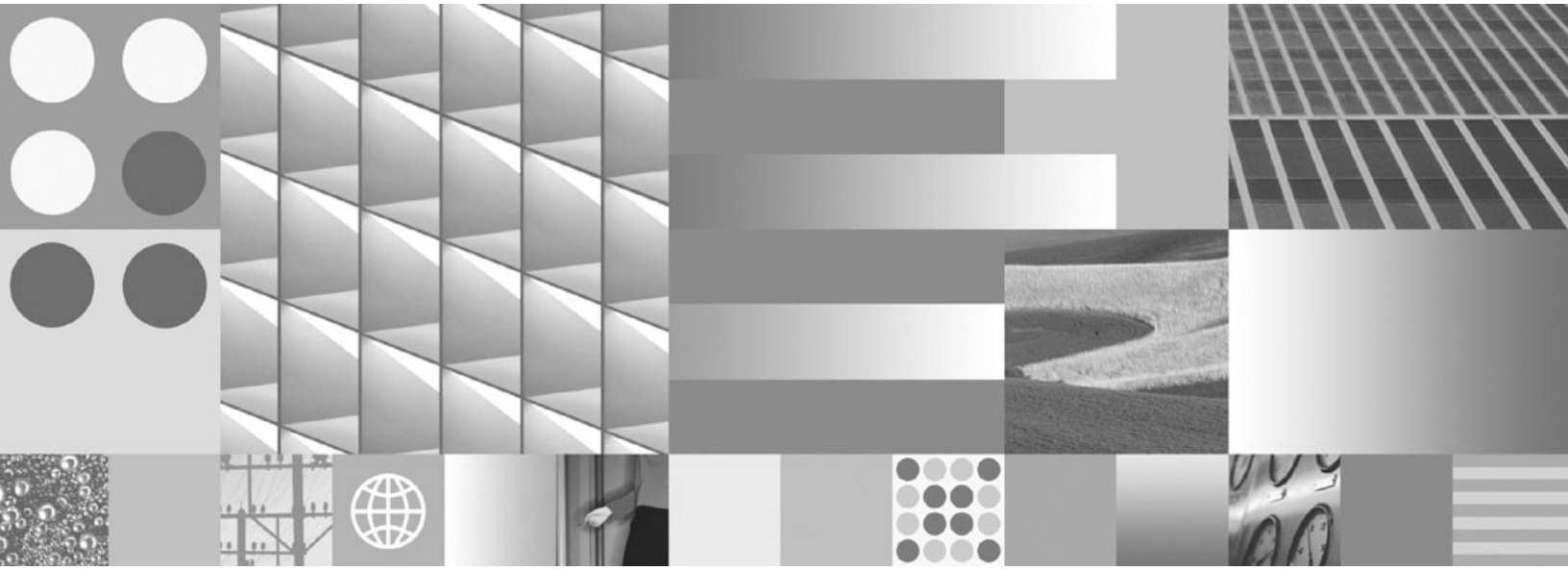
IBM OmniFind Enterprise Edition

8.4 版

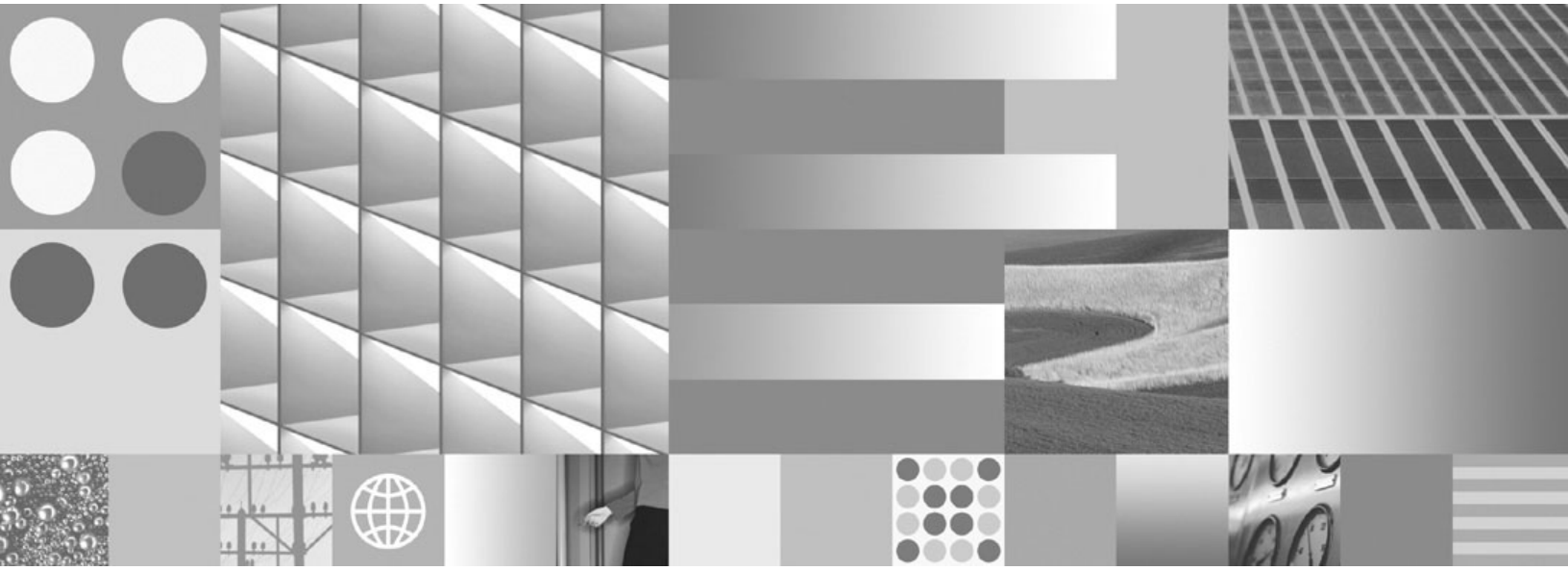


文字分析整合

SC40-2071-01

IBM OmniFind Enterprise Edition

8.4 版



文字分析整合

SC40-2071-01

注意事項

使用此資訊及其支援的產品之前，請務必先閱讀第 101 頁的『注意事項與商標』中的一般資訊。

第二版 (2006 年 11 月)

您可以線上訂購 IBM 出版品，或可以透過當地的 IBM 業務代表來訂購：

- 若要線上訂購出版品，請造訪「IBM 出版品中心 (IBM Publication Center)」：www.ibm.com/shop/publications/order。
- 若要尋找當地的 IBM 業務代表，請造訪「IBM 全球聯絡站名錄 (IBM Directory of Worldwide Contacts)」：www.ibm.com/planetwide。

當您傳送資訊給 IBM 時，即授權予 IBM，IBM 得以其認爲適當的方式來使用或分送資訊，而無需對您負任何責任。

目錄

語意搜尋的語言支援

企業搜尋針對大部分的印歐語言及亞洲語言 (包括日文) 的純文字文件，提供語言搜尋支援。

您可以使用語言支援來提高搜尋結果的品質。

語言處理的執行分成兩個階段：處理要新增至索引的文件階段，以及使用者輸入搜尋查詢階段。

企業搜尋僅包括精細或基本的語言功能，用來判定輸入文件的語言，以及將文件輸入串流分段成單字或記號。

如果您知道搜尋會主要限於基本關鍵字搜尋或使用文件結構的原生 XML 搜尋，則企業搜尋所含的語言處理作業會適當地涵蓋您的需要。

純文字文件中的大部分資訊都是非結構化的，因爲不易取得資訊的意義，所以會很難有效地使用這些資訊。

搜尋關鍵字很簡單，然而如果您要在文件中搜尋純單字以外的項目，則不一定能滿足需求，如下面範例所示：

- 在協同作業的情況下，資訊不一定會明確地標示出來，例如，電子郵件中的位址或電話號碼。實際上，可能根本不會使用*電話號碼* 一詞。而電子郵件中可能會包含「您可以撥打 555-641-1805 來聯絡我」之類的詞組。使用者通常並不知道他們要搜索的資訊在文件中呈現的方式，當尋找名爲 Barbara 的某人電話號碼時，理論上就會想要輸入類似於「Barbara 電話號碼」的查詢。然而，因爲*電話號碼* 一字並未出現在文件中，所以這個查詢不會成功。
- 在競爭情報中，文件提到競爭者及其提供的貨品，或競爭者的網站在過去三個月從銷售某一產品組移至銷售另一產品組。在此情況下，使用者可能會輸入類似於「Smith 及 Co. 貨品」或「Smith 及 Co. 貨品 2004 年 11 月至 2005 年 1 月」的查詢。在第一個查詢中，*貨品*一詞代表產品或產品範圍，然而因爲尋找的是*貨品*一詞，所以不會傳回 Smith & Co. 所提供的產品。此情況同樣適用於包括特定時段的查詢。使用關鍵字搜尋幾乎不可能查詢到時段。
- 在客戶關係管理中，文件可能會提到舊金山區修車廠的汽車煞車問題。修車廠報告說明「由於油壓洩露而調整煞車皮」之類的狀況。想要查詢詳細資訊的使用者可能會輸入「北舊金山修車廠的煞車問題」之類的查詢。然而，因爲*煞車問題* 或*修車廠* 一詞不會出現在報告中，所以此查詢不會傳回關於「由於油壓洩露而調整煞車皮」的任何報告。此外，這些報告可能只提到修車廠的街名及區名，而不是包括城市名稱「舊金山」在內的完整地址。
- 在研究中，文件說明以各種商標在市場上廣泛銷售的特定藥品，以及它與同一段落中所提到的至少一種疾病之間的關係。一般的使用者可能會輸入藥品的其中一個常見字詞進行查詢，希望取得包括症狀在內之各種疾病的詳細說明。然而，因爲文件中不一定會使用常見的字詞，並且這些文件通常不會提到*疾病* 一詞，而只會提到疾病本身的名稱，所以該查詢可能不會傳回令人滿意的文件。

在這些範例中，在現今存在的廣大資源來源集合中搜尋您需要的項目，這是一項新挑戰，需要比企業搜尋所提供的斷詞法層次及定義檔型分析更精確的分析。大部分需要的資訊都不會在原始文件中明確地標示或標記。而是必須分析文件內容，才能辨識和尋找所關注的概念，例如，人員、組織、位置、機能及產品之類的具名實體，以及這些實體之間的可能關係。

您想要在純文字文件中探索及擷取的資訊是使用者與網域的專用資訊。爲了協助您設計及開發專屬的分析演算法，IBM® 會提供 IBM Unstructured Information Management Architecture (UIMA)，它是一種軟體架構，可協助您在企業搜尋中建置進階分析功能，以便在文件集合中尋找所關注的資訊。

相關概念

第 3 頁的『自訂文字分析整合』
使用「非結構化資訊管理架構 (UIMA)」在企業搜尋外部建置自訂分析之後，您可以在企業搜尋中使用企業搜尋管理主控台整合分析邏輯。

第 3 頁的『文字分析程序中使用的基礎概念』
文字分析處理作業中使用的基礎概念包括註解程式、分析結果、特性結構、類型、類型系統、註解及共用分析結構。

自訂文字分析整合

使用「非結構化資訊管理架構 (UIMA)」在企業搜尋外部建置自訂分析之後，您可以在企業搜尋中使用企業搜尋管理主控台整合分析邏輯。

UIMA 是一種開放式平台，可以針對每種不同概念的分析功能來識別元件，並確保可以輕易地重覆使用及結合這些元件。

進階語言分析可以包括許多不同分析作業的組合。分析是從語言偵測與斷詞法開始，然後是詞性識別，接著執行深度文法剖析。舉例來說，最後一個作業包括識別特定化學物質與特定徵兆外觀之間的關係。分析程序中的每一個步驟都視前一個步驟的結果而定。

每一個步驟的分析邏輯都包含在*註解程式* 中。註解程式會結合以形成一連串的處理作業，重複執行集合中的每一個文件，來探索新的資訊並儲存此資訊以供下游處理作業使用。

負責探索及代表純文字文件中分析內容的註解程式，內含在*分析引擎*中 (UIMA 的中心概念)。分析引擎可以包含單一註解程式，或者可以是許多引擎的組合，而每一個引擎又都包含註解程式。

UIMA 僅提供基本建置區塊，供您建立、測試及部署自己的分析引擎使用。它不會以預先配置的分析引擎形式，提供可讓您在 UIMA 環境中部署的任何語言分析功能。然而，企業搜尋中套用的語言處理作業可作爲一組註解程式在 UIMA 中使用。

若要使用 UIMA，您必須安裝「UIMA 軟體開發套件」。開發工具箱可在 IBM developerWorks® 中取得。請造訪 WebSphere® Information Integrator 區域，以取得 http://www.ibm.com/developerworks/db2/zones/db2ii/ 上的資訊。 「UIMA 軟體開發套件 (SDK)」包括 UIMA 架構的 Java™ 實作，以供實作、說明、組合及部署 UIMA 元件。

UIMA SDK 還提供一組工具與公用程式，以便在 Eclipse 型開發環境 (Eclipse 外掛程式) 中使用 UIMA。如需 Eclipse 的相關資訊，請參閱 www.eclipse.org，如需如何在「Eclipse 互動式開發環境」中安裝「UIMA 軟體開發套件」的相關指示，請參閱 UIMA 文件。

相關概念

第 1 頁的『語意搜尋的語言支援』
企業搜尋針對大部分的印歐語言及亞洲語言 (包括日文) 的純文字文件，提供語言搜尋支援。

『文字分析程序中使用的基礎概念』
文字分析處理作業中使用的基礎概念包括註解程式、分析結果、特性結構、類型、類型系統、註解及共用分析結構。

文字分析程序中使用的基礎概念

文字分析處理作業中使用的基礎概念包括註解程式、分析結果、特性結構、類型、類型系統、註解及共用分析結構。

註解程式 包含分析文件的邏輯，並探索及記錄文件相關的敘述性資料，將該資料作爲文件的整體 (稱爲文件中間資料) 或部分邏輯。此敘述性資料被稱爲*分析結果*。分析結果會註解純文字文件中所有連續的子字串 (也稱爲跨距)。理論上，分析結果會對應於您要搜尋的資訊。

特性結構 是代表分析結果的基礎資料結構。特性結構是屬性值結構。每一個特性結構都屬於某一*類型*，而每一個類型都具有一組指定的有效特性或屬性 (內容)，非常類似 Java 類別。特性含有範圍類型，指出特性必須具備的值類型，如 String。

例如，橫跨 "James Matthew Bloggs" 的文字可能由類型 `Person`、特性爲 `personName`、`age`、`nationality` 及 `profession` 的註解橫跨。

類型系統 定義可能會在文件中發現之物件 (特性結構) 的類型。類型系統根據類型及特性 (屬性)，定義所有可能的特性結構，非常類似 Java 的類別階層。您可以在類型系統中，定義任意數目的不同類型。類型系統和網域及應用程式攸關。

大部分的文字分析註解程式以*註解* 的形式產生分析結果。註解是一種特殊類型的特性結構，主要用於語言分析處理。註解會橫跨或涵蓋輸入文字的片段，且是根據其在輸入文字中的開頭及結束位置定義的。

例如，辨識貨幣表示式的註解程式針對文字 "100.55 US Dollars" 建立了類型 `monetaryExpression` 的註解，並將特性 `currencySymbol` 設爲 "$" 以替代該文字。

所有註解程式都位於 UIMA 模型，並將資料儲存在特性結構中。

所有特性結構都會顯示在稱爲*共用分析結構*的中央資料結構中。您可以利用共用分析結構來處理所有資料交換。

共用分析結構包含下列物件：

- 純文字文件
- 類型系統說明，指出類型、次類型及其特性
- 分析結果，說明文件或文件的區域
- 索引儲存庫，支援分析結果的存取和疊代

相關概念

第 1 頁的『語意搜尋的語言支援』
企業搜尋針對大部分的印歐語言及亞洲語言 (包括日文) 的純文字文件，提供語言搜尋支援。

第 3 頁的『自訂文字分析整合』
使用「非結構化資訊管理架構 (UIMA)」在企業搜尋外部建置自訂分析之後，您可以在企業搜尋中使用企業搜尋管理主控台整合分析邏輯。

文字分析演算法

「UIMA 軟體開發套件」包括 API 及工具，您可以用來建立註解程式 (分析演算法，包括類型系統說明)，並在分析引擎中嵌入這些註解程式。

UIMA 文件包括教學指導樣式手冊，可協助您建置這些元件。「軟體開發套件」包括測試及檢視結果的公用程式，以及索引分析結果的小型語意搜尋引擎。您也可以針對儲存在索引中的資訊執行更多進階語意搜尋。

因爲「UIMA 軟體開發套件」不提供任何預先配置的註解程式，且您使用 UIMA 開發然後在企業搜尋中整合的所有自訂註解程式都是根據企業搜尋基礎註解程式的結果建置的，所以您可以在 UIMA 環境中使用基礎註解程式資料包。如需在 UIMA 環境中執行自訂文字分析演算法之前，如何併入語言偵測及記號化功能的相關資訊，請參閱 UIMA 文件。

使用「UIMA 軟體開發套件」開發及測試您的分析引擎之後，必須建立 PEAR (「處理程序引擎保存檔」) 檔案，以在企業搜尋中針對文件集合執行演算法。這個保存檔包括在企業搜尋中部署自訂分析功能作爲分析引擎所需的全部資源。如需如何建立保存檔的相關資訊，請參閱「軟體開發套件」提供的 UIMA 文件。

您建立用來上載企業搜尋的保存檔必須僅包含您的自訂分析邏輯。因爲在企業搜尋中，基礎註解程式始終會在任何自訂分析之前執行，所以即使自訂分析邏輯根據基礎註解程式結果建置，該保存檔也不能包含任何企業搜尋基礎註解程式。

若要瞭解如何在企業搜尋中配置及部署語意搜尋解決方案，請執行 http://www.ibm.com/developerworks/db2/zones/db2ii/ 所述的教學指導。教學指導會引導您完成在企業搜尋中部署自訂文字分析演算法的步驟，並顯示如何在查詢中使用分析結果來增進搜尋結果。

相關工作

第 6 頁的『使用 UIMA 中的企業搜尋基本註解程式』
您可以使用企業搜尋基本註解程式套件中的註解程式，在「UIMA 軟體開發套件 (SDK)」中開發新的註解程式，並將分析結果對映至 JDBC 表格。

自訂分析整合的工作流程

您可以使用「UIMA 軟體開發套件」來建立及測試自訂文字分析演算法，然後對企業搜尋中的文件集合部署及執行這些演算法。

若要開發分析演算法並在企業搜尋中對其進行整合，請：

1. 計畫及設計
   a. 決定您要搜尋的資訊。您要擷取哪些文件？ 每一個特定搜尋作業需要哪些概念及關係？例如，若要加強製藥公司內部網站的一般用途搜尋，可能需要產品及員工名稱，而在研究及開發區的人員需要使用藥品名稱的變體，並查看藥品-原因-療法關係。
   b. 指定在您要搜尋的文件中擷取資訊所需的文字分析類型。
   c. 如果集合含有 XML 文件，請決定是否要在解決方案中利用 XML 標記。在企業搜尋中，您可以使用下列兩種方式來利用 XML 標記：
      - 如果您可以在自訂分析中使用 XML 標記 (例如，文件包含有助於彙總或分類註解程式的 `<summary>` 或 `<topic>` 元素)，請建立 XML 元素到共用分析結構的對映檔。
      - 如果您要在查詢中依照 XML 標記在文件中的顯示方式來使用它，則必須啓用原生 XML 對映。
   d. 決定您要利用語意搜尋存取共用分析結構中儲存的哪些文字分析結果資訊。建立共用分析結構到索引的對映檔。
   e. 決定是否要在關聯式資料庫儲存分析結果，例如，利用報告及資料採礦應用程式來探查趨勢及關聯。建立共用分析結構到資料庫的對映檔。

f. 設計語意搜尋應用程式。決定搜尋使用者如何使用語意搜尋的其他功能。設計使用者介面。

2. 開發：「UIMA 軟體開發套件」活動
   a. 定義個別的分析步驟。
   b. 說明對映及分析演算法的類型系統。
   c. 利用「UIMA 軟體開發套件」，開發每一個分析步驟的分析演算法 (註解程式)，並在分析引擎中嵌入註解程式。利用企業搜尋基本註解程式資料包中的基本功能 (語言識別及分段)，建置任何自訂分析。
   d. 在 UIMA 中測試分析演算法之後，將分析引擎封裝爲 PEAR 檔 (「處理程序引擎保存檔」)。保存檔只能包含您的分析演算法，而不包含基本企業搜尋語言功能。

      當您設計文字分析解決方案時，其可以包括在多個 PEAR 檔案中提供的數個分析模組。UIMA 提供將兩個或多個 PEAR 檔案併入單一 PEAR 檔案的方法，您可以在企業搜尋中上載及執行這些檔案。合併 PEAR 檔的機能可確保沒有命名衝突、正確合併輸入及輸出功能，以及當註解程式描述子中的合併參數具有相同的名稱時不會置換參數。如需如何合併 PEAR 檔的相關指示，請參閱 UIMA 文件。

3. 部署：企業搜尋活動
   a. 將處理作業引擎保存檔 (.pear) 上載至企業搜尋。提供文字分析元件的名稱，以便能在企業搜尋中參照它。
   b. 建立一或多個文件集合與文字分析元件的關聯性。
   c. 如果適用，請針對每一個集合，上載及選取您針對自訂分析定義之 XML 元素到共用分析結構的對映。
   d. 如果適用，請針對每一個集合，上載及選取您針對自訂分析定義之共用分析結構到資料庫的對映。
   e. 針對每一個集合，上載及選取您針對語意搜尋定義之共用分析結構到索引的對映。
   f. 必要的話，請設定自訂語意搜尋應用程式，例如，將瀏覽器型搜尋使用者介面部署到應用程式伺服器。
   g. 搜索、剖析及檢索語意搜尋集合中的文件，就像您在關鍵字型集合中所做的一樣。

相關工作

『使用 UIMA 中的企業搜尋基本註解程式』
您可以使用企業搜尋基本註解程式套件中的註解程式，在「UIMA 軟體開發套件 (SDK)」中開發新的註解程式，並將分析結果對映至 JDBC 表格。

使用 UIMA 中的企業搜尋基本註解程式

您可以使用企業搜尋基本註解程式套件中的註解程式，在「UIMA 軟體開發套件 (SDK)」中開發新的註解程式，並將分析結果對映至 JDBC 表格。

一組基本註解程式包括：

- **語言 ID 註解程式**

  偵測文件的語言。如需功能及配置參數，請參閱描述子檔案 jlangid.xml。

- **FROST 定義檔查閱註解程式**

  依據 IBM LanguageWare 定義檔，提供分段及句子偵測。若爲記號，則會產生其他語言資訊，例如，基礎詞形或詞形。如需功能及配置參數，請參閱描述子檔案 jfrost.xml。

- **空格記號器**

  可以對所有歐洲語言文件執行以空格爲基礎的記號化或其他空格區隔的 Script。此外，註解程式也可以對下列文字 Script 執行 n-gram 記號化：阿拉伯文、漢語、希伯萊文、平假名、片假名、寮文、蒙古文、泰文、YI，以及韓文。此清單包括所有主要亞洲文字 Script，並表示註解程式支援日文、中文及韓文。

  如需功能及配置參數，請參閱描述子檔案 jtok.xml。

- **正規表示式註解程式**

  根據正規表示式，偵測純文字文件中資訊的實體或跨距。您可以自訂正規表示式註解程式，以透過定義您自己的規則來偵測所需要的文字實體。在純文字文件中偵測電話號碼、URL 及電子郵件位址的範例正規表示式註解程式內含在基本註解程式資料包中。

- **資料庫消費者的共用分析結構**

  資料庫消費者的共用分析結構會將特定的文字分析結果輸入關聯式資料庫。

企業搜尋基本註解程式套件是一個 zip 格式壓縮檔案，其中包含基本文字分析註解程式、正規表示式註解程式及資料庫消費者的共用分析結構。「語言 ID」註解程式、FROST 定義檔查閱註解程式及空格記號器都是基本文字分析註解程式，當在企業搜尋中剖析文件時，其始終在所有自訂文字分析之前執行。

因爲基本文字分析註解程式始終在企業搜尋中的所有自訂文字分析之前執行，且所有自訂文字分析都以基本註解程式的輸出爲基礎，所以當您開發及測試自訂註解程式時，可以將這些註解程式用於 UIMA 環境。

正規表示式註解程式及共用分析結構到資料庫的消費者，是您在配置文字處理作業選項時，可以在企業搜尋管理主控台上選取的其他選項。您也可以將其用於 UIMA。針對正規表示式註解程式的進階自訂，建議您使用提供的 UIMA SDK 工具來自訂註解程式。

若要在 UIMA 中執行其中任何的註解程式，您必須安裝「UIMA 體開發套件 (SDK)」。其位於 IBM developerWorks 網站，網址爲 http://www.ibm.com/developerworks/db2/zones/db2ii/。

若要在您的 UIMA SDK 安裝中安裝註解程式套件，請：

1. 在企業搜尋 (WebSphere Information Integrator OmniFind™ Edition) 安裝的 *ES_INSTALL_ROOT*/packages/uima 目錄中，註解程式資料包 OF_base_annotators.zip。
2. 將壓縮檔複製到 UIMA SDK 安裝的根目錄。
3. 解壓縮 zip 檔，將企業搜尋基本註解程式檔案加入 UIMA SDK 安裝的指定目錄結構。這樣會覆寫檔案 tt_core_typesystem.xml。如果您想要保留此檔案的舊版本，請先儲存壓縮檔，然後再解壓縮它。

4. 設定類別路徑、開啓 bin 目錄中的 setUIMAClasspath Script，並在啓動 OFAnnotEnv Script 的 Script 結尾新增一行。
5. 如果您要在 UIMA 中使用任何自訂或企業搜尋特定類型，請參閱關於如何定義這些類型的 UIMA SDK 文件。

安裝基本註解程式套件之後，您可以在目錄 *UIMA_SDK_INSTALL*/docs/examples/descriptors/analysis_engine 中找到註解程式描述子檔案。檔案 of_tokenization.xml 會依照其在企業搜尋內的使用順序，列示基本文字分析註解程式 (「語言 ID」註解程式、FROST 定義檔查閱註解程式及空格記號器)。

描述子檔案所含的配置值與企業搜尋中使用的值相同。您可以在 UIMA SDK 中變更值以進行除錯。然而，請勿在企業搜尋系統中變更這些描述子檔案。變更這些檔案可能會造成系統不穩定或效能問題。

企業搜尋基本註解程式資料包只含有處理英文文件所需的定義檔。如果您要在開發環境中處理其他語言，請請遵循下列步驟：

1. 在企業搜尋安裝的 *ES_INSTALL_ROOT*/configurations/parserservice/jediidata/frost/resources 中，尋找企業搜尋定義檔。
2. 將定義檔的內容複製到本端 UIMA SDK 安裝的 *UIMA_SDK_INSTALL*/data/frost/resources 。

若要驗證註解程式資料包是否已順利安裝：

1. 在下列目錄中開啓「共用分析結構 (CAS) 視覺除錯器 (CVD)」：*UIMA_SDK_INSTALL*/bin/cvd[.bat/.sh]。
2. 按一下**執行 → 載入 TAE**。
3. 在 *UIMA_SDK_INSTALL*/docs/examples/descriptors/analysis_engine 目錄中選取文字分析引擎指定元檔案 of_tokenization.xml。
4. 載入範例文件，並執行文字分析引擎。您會在 CVD 中看到類型 `uima.tt.TokenAnnotation` 的註解。

如果您在開發環境中執行自訂註解程式之前執行任何基本文字分析註解程式，且您的自訂註解程式使用由基本文字分析定義的類型，則會在自訂註解程式指定元的類型系統區段中併入檔案 `tt_core_typesystem` 的參照。`tt_core_typesystem` 檔案位於 *UIMA_SDK_INSTALL*/docs/examples/descriptors/analysis_engine 目錄。若需如何併入描述子檔案參照的範例，請參閱 analysis_engine 目錄中的 jtok.xml 檔案。

相關工作

第 11 頁的『檢視基本註解程式及自訂文字分析結果』
若要檢視在剖析之後由企業搜尋中任何註解程式所產生的分析結果，您必須更新文件集合屬性，以產生儲存在共用分析結構中之分析結果的可讀 XML 版本。

第 74 頁的『啓用使用正規表示式註解程式的簡易語意搜尋』
若要使用同義字啓用簡易語意搜尋，您必須將正規表示式註解程式、共用分析結構到索引的對映檔及範例同義字定義檔新增至企業搜尋系統，並建立這些資源與集合的關聯性。

第 9 頁的『在 UIMA 中使用資料庫消費者的共用分析結構』
在 UIMA 中使用資料庫消費者的共用分析結構之前，您必須變更消費者描述子檔案，並將共用分析結構寫入資料庫對映檔中。

第 10 頁的『在 UIMA 中使用正規表示式註解程式』
使用正規表示式註解程式偵測純文字文件中資訊的實體或單位。您可以自訂主體網域的註解程式，以滿足您的搜尋需要。

在 UIMA 中使用資料庫消費者的共用分析結構

在 UIMA 中使用資料庫消費者的共用分析結構之前，您必須變更消費者描述子檔案，並將共用分析結構寫入資料庫對映檔中。

您必須首先執行下列作業，才能在 UIMA 環境中執行資料庫消費者的共用分析結構：

1. 開啓 *UIMA_SDK_INSTALL*/docs/examples/descriptors/cas_consumer 中的 XML 描述子檔案 cas2jdbc.xml。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。
2. 修改參數 **mappingFile** 以包括絕對路徑，其爲共用分析結構到資料庫的對映檔位置，例如，D:\temp\MyMapping.xml
3. 修改參數 **docMetadata_Type** 以指定 UIMA 類型，可以從此類型擷取所有內建特性的中間資料，例如，uima.tcas.DocumentAnnotation。
4. 修改參數 **docId_Feature** 以包括中間資料類型的特性或特性路徑，可以從中擷取文件的數字 ID (整數類型)。需要 ID 的所有內建特性 (例如 `docId()`、`uniqueId()`、`objectId()` 及 `fsId()`) 都需要此項目。
5. 因爲參數 **encryptionClass** 僅用於企業搜尋中，所以請不要設定它，以容許資料庫消費者的共用分析結構使用已加密的對映檔。
6. 儲存檔案。
7. 將 EMF 程式庫檔案 (common.jar、ecore.jar 及 ecore.xmi.jar) 從企業搜尋安裝的 lib 目錄複製至 UIMA 安裝的 lib 目錄。 cc_cas2jdbc.jar 已經位於 UIMA 安裝的 lib 目錄中。
8. 建立共用分析結果到資料庫的對映檔，其定義要將哪些文字分析結構儲存在資料庫中。 您可以用位於 *UIMA_SDK_INSTALL*/docs/examples/descriptors/cas_consumer 的對映檔 sampleMapping.xml 作爲範例，來建立您自己的對映檔。

   使用位於 *UIMA_SDK_INSTALL*/docs/examples/descriptors/cas_consumer 且名爲 CasToJDBCMapping.xsd 的 XML 綱目檔案，來驗證共用分析結構到資料庫的對映檔。由於效能原因，資料庫消費者的共用分析結構不會驗證對映檔，所以您必須親自驗證該檔案。

UIMA 文件說明如何在 UIMA 中執行消費者。

下列範例顯示必須如何在描述子中定義必要參數：

```
 ...
 <nameValuePair>
<name>mappingFile</name>
  <value>
<string>D:/temp/MyMapping.xml</string>
  </value>
 </nameValuePair>
<nameValuePair>
<name>docMetadata_Type</name>
  <value>
<string>uima.tcas.DocumentAnnotation</string>
  </value>
 </nameValuePair>
<nameValuePair>
```

```
<name>docId_Feature</name>
  <value>
<string>end</string>
  </value>
 </nameValuePair>

...
```

表格會依照配置參數在描述子檔案中出現的順序來顯示它們，並指出哪些是必要參數：

表 1. 資料庫消費者描述子檔案之共用分析結構中的配置參數

參數	說明	必要
mappingFile	共用分析結構到資料庫之對映檔的絕對路徑，例如，D:/temp/sample.xml。在 Windows® 系統上，使用"/"作爲路徑分隔字元。	true
encryptionClass	請不要設定此參數，因爲它僅用於企業搜尋中，以容許資料庫消費者的共用分析結構使用已加密的對映檔。	false
docMetadata_Type	從中擷取內建特性所有中間資料的 UIMA 類型。	true
docId_Feature	從中擷取文件數字 ID 之中間資料類型上的特性或特性路徑。其類型必須是整數，而且需要 ID 的所有內建特性 (例如 `uniqeId()`、`objectId()` 及 `fsId()`) 都需要此項目。	true
docUri_Feature	從中擷取文件 URI 之中間資料類型上的特性或特性路徑。它必須是字串類型。	false
IsCompleted_Feature	中間資料類型上的特性或特性路徑，該資料類型會標示是否跨越數個共用分析結構將現行文件分段。	false
chunkNumber_Feature	表示現行片段之後續號碼的中間資料類型上的特性或特性路徑。	false

在 UIMA 中使用正規表示式註解程式

使用正規表示式註解程式偵測純文字文件中資訊的實體或單位。您可以自訂主體網域的註解程式，以滿足您的搜尋需要。

若要執行偵測電話號碼、URL 及電子郵件位址的範例正規表示式註解程式，或使用範例註解程式作爲在 UIMA 環境中建立您自訂版本之正規表示式註解程式的基準，您需要：

1. *UIMA_SDK_INSTALL*/docs/examples/descriptors/analysis_engine 目錄中的正規表示式註解程式描述子。
2. *UIMA_SDK_INSTALL*/docs/examples/regex 目錄中的範例規則集及類型系統說明。

3. 範例規則集已套用於 *UIMA_SDK_INSTALL*/docs/data 目錄的範例文字檔，稱爲 of_sample_regex.txt。

UIMA 文件說明如何在 UIMA 中執行註解程式。

檢視基本註解程式及自訂文字分析結果

若要檢視在剖析之後由企業搜尋中任何註解程式所產生的分析結果，您必須更新文件集合屬性，以產生儲存在共用分析結構中之分析結果的可讀 XML 版本。

關於本作業

使用儲存在共用分析結構中之註解程式分析結果的 XML 序列化，您可以：

- 在處理基本註解程式之前，檢視剖析後的結果。
- 檢視剖析及記號化後的結果 (執行企業搜尋基本註解程式)。這可協助您決定要開發之任何自訂分析的輸入資料結構，其始終在基本註解程式之後執行。
- 決定在完整集合上執行分析之前，檢視及驗證在企業搜尋中更小文件集合上執行之自訂分析的結果，以進行測試。

XML 序列化會產生兩個結果集：

- 剖析後的結果。這些結果包括欄位對映及文件中間資料。
- 剖析及記號化後的結果 (如果選取的話) 會自訂文字分析。這些結果包括所有產生的記號及註解。

程序

若要產生分析結果的可讀 XML 版本，請：

1. 開啓 *ES_NODE_ROOT*/master_config/<CollectionID>.parserdriver 中的檔案 collection.properties，然後開始剖析集合中的文件。
2. 若要檢視剖析後的結果，請將下列行新增至 collection.properties 檔：
   `trevi.parser.dumpXCas=<your_dump_directory>`

   您的傾出目錄必須已存在。

   a. 選取您要的輸出類型。 輸出始終包括用於剖析結果的類型系統說明，稱爲 OmniFindParserTypeSystem.xml。新增下列其中一行：

      - 若要檢視最後處理之 25 個檔案的輸出，請新增 `trevi.parser.maxXCasFileCount=25`。

        雖然不建議您將檔案的數目設的過高，但是您仍然可以自行決定此值。

        請注意，在達到最大緩衝區大小之後，會不斷改寫檔案輸出緩衝區。這也表示，不需要最後處理編號最大的文件。

        輸出包括下列檔案：OmniFindParserXCasDump1.xml、接著是 OmniFindParserXCasDump2.xml，依此類推，直至列出 25 個檔案爲止。
      - 若要檢視特定文件的輸出，請新增文件 URI `trevi.parser.xCasURI.1=file://home/test/file1.txt`。

您可以新增任何數目的文件，然而，文件必須從 1 開始以連續數字按遞增次序編號。例如，第二個文件是 `trevi.parser.xCasURI.2=file://home/test/file2.txt`，則第三個文件是 `trevi.parser.xCasURI.3=file://home/test/file3.txt`

輸出包括下列檔案：
OmniFindParserXCasDumpURI_1.xml、OmniFindParserXCasDumpURI_2.xml 等您列示的所有檔案名稱。

3. 若要檢視記號化後的結果，請新增下列行：
`trevi.tokenizer.dumpXCas=<your_dump_directory>`

此外，您的傾出目錄必須已存在。

a. 選取您要的輸出類型。 已建立的輸出始終包括用於記號化的類型系統說明，以及稱爲 OmniFindTypeSystem.xml 的文字分析結果。新增下列其中一行：

- 若要檢視最後處理之 25 個檔案的輸出，請新增 `trevi.tokenizer.maxXCasFileCount=25`。

  雖然不建議您將檔案的數目設的過高，但是您仍然可以自行決定此值。

  請注意，在達到最大緩衝區大小之後，會不斷改寫檔案輸出緩衝區。這也表示，不需要最後處理編號最大的文件。

  輸出包括下列檔案：OmniFindXCasDump1.xml、OmniFindXCasDump2.xml，依此類推，直至列出 25 個檔案爲止。

- 若要檢視特定文件的輸出，請新增文件 URI `trevi.tokenizer.xCasURI.1=file://home/test/file1.txt`。

  您可以新增任何數目的文件，然而，文件必須從 1 開始以連續數字按遞增次序編號。例如，第二個文件是 `trevi.tokenizer.xCasURI.2=file://home/test/file2.txt`，則第三個文件是 `trevi.tokenizer.xCasURI.3=file://home/test/file3.txt`

  輸出包括下列檔案：
  OmniFindXCasDumpURI_1.xml、OmniFindXCasDumpURI_2.xml 等您列示的所有檔案名稱。

在企業搜尋中，您可以使用「XCAS 註解檢視器」來檢視 XML 檔的內容。執行位於 ES_INSTALL_ROOT/bin 目錄中的 `xcasAnnotationViewer` Script 檔，來啓動「XCAS 註解檢視器」。系統會提示您：

- 剖析或記號化之後放置結果的傾出目錄
- 描述子檔案，OmniFindParserTypeSystem.xml (適用於剖析器結果) 或 OmniFindTypeSystem.xml (適用於記號化及分析結果)，同樣位於您的傾出目錄中。

選取清單中的文件會顯示文件的分析結果。按一下文件中強調顯示的註解，會顯示註解的詳細資料。

類型系統說明

類型系統定義物件類型及其屬性 (或特性)，您可以在共用分析結構中實例化這些屬性。

每一個分析引擎都有自己的類型系統說明，其說明分析引擎中註解程式的輸入需求和輸出類型。類型系統說明是應用程式網域專用的。

類型系統包括類型定義、類型屬性及類型的單一繼承階層。共用分析結構必須符合特定的類型系統。

定義在類型系統說明中的類型及特性還必須用於與文件分析相關聯的所有對映檔，包括 XML 元素到共用分析結構的對映檔、共用分析結構到索引的對映檔，以及共用分析結構到資料庫的對映檔。

註解程式的類型系統說明可以是註解程式描述子的一部分，或可以內含於不同的類型系統描述子檔案中。有時，它是相同分析引擎所含的另一個註解程式的描述子。

當您在 UIMA 環境中完成開發及測試分析引擎時，您所建立並上載至企業搜尋的保存檔 (.pear 檔) 就會包含分析邏輯檔及類型系統說明。

企業搜尋基本註解程式使用三種類型系統說明：一個是始終包括的核心類型系統說明，您可以選擇性地啓動其他兩個種類型系統說明，以將文件集合基本分析處理作業變更爲進階分析模式。您是否需要包括其中一個或這兩個延伸類型系統說明，視您在基本分析處理作業期間要併入的其他文字分析處理作業結果而定。

您可以藉由包括一或兩個延伸類型系統來啓用進階分析模式。在進階分析模式中，其他分析特性在基本分析處理作業期間可用，並儲存在共用分析結構中。例如，如果您需要記號的相關資訊 (更多特性資訊) ，如記號的所有可能詞形，或者詞形是否爲停用字、詞形的詞性或型態處理作業及日文的特殊特性，您需要啓動進階分析模式。

相關工作

『從基本分析模式變更爲進階分析模式』
若要將由企業搜尋基本註解程式執行的文件集合處理作業從基本分析模式變更爲進階分析模式，您必須併入進階分析模式的類型系統說明。

相關參考

第 14 頁的『定義於企業搜尋的類型及特性』
定義於企業搜尋的類型系統包含文件中間資料處理及基本語言分析。

從基本分析模式變更為進階分析模式

若要將由企業搜尋基本註解程式執行的文件集合處理作業從基本分析模式變更爲進階分析模式，您必須併入進階分析模式的類型系統說明。

限制

您可以選取下列兩種類型系統說明來啓動進階分析模式：

- `tt_extension_typesystem` 說明，其包括詞形更詳細的詞彙類型特性資訊。
- `dlt_extension_typesystem` 說明，其包括其他型態特性及特殊詞彙類型。

程序

若要將基本集合處理作業變更爲進階分析模式，請：

1. 開啓 *ES_NODE_ROOT*/master_config/*CollectionID*.parserdriver/specifiers 目錄中的檔案 tt_core_typesystem.xml。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。

2. 移除 <imports> 區段中 <import> 元素周圍的註解標示，以併入其中一個或這兩個延伸類型系統說明檔案。

```
<imports>
<!-- imports the tt_extension_typsystem for advanced analysis -->
<!-- <import location="tt_extension_typesystem.xml"/>-->
<!-- imports dlt extension typesystem -->
<!-- <import location="dlt_extension_typesystem.xml"/> -->
</imports>
```

3. 開啓兩個描述子檔案 jfrost.xml 及 jfrost_ngram.xml，並修改 <outputs> 元素的內容，以併入您在分析期間要併入的類型 (在 <type> 元素中) 及特性 (在 <feature> 元素中)，該類型及特性列示在 <capabilities> 區段的 <description> 元素中。 儲存變更。
4. 開啓描述子檔案 jtok.xml，並修改 <outputs> 元素的內容，以併入您在分析期間要併入的特性 (在 <feature> 元素中)，該特性列示在 <capabilities> 區段的 <description> 元素中。 儲存變更。
5. 開啓描述子檔案 es_tok_no_stw.xml，並還請在此處修改 <outputs> 元素的內容，以併入您在分析期間要併入的特性 (在 <feature> 元素中)，該特性列示在 <capabilities> 區段的 <description> 元素中。 儲存變更。
6. 變更爲進階分析模式後，您必須重新剖析文件集合。

相關概念

第 12 頁的『類型系統說明』
類型系統定義物件類型及其屬性 (或特性)，您可以在共用分析結構中實例化這些屬性。

相關參考

『定義於企業搜尋的類型及特性』
定義於企業搜尋的類型系統包含文件中間資料處理及基本語言分析。

定義於企業搜尋的類型及特性

定義於企業搜尋的類型系統包含文件中間資料處理及基本語言分析。

企業搜尋中使用的類型分別定義於三個類型系統說明檔中，從包含核心類型 (始終用於所有基本語言分析) 的類型系統說明檔開始，隨後是定義進階語言功能 (通常僅用於進階分析模式) 的類型系統說明。

索引文件時，一定會發生文件語言識別及斷詞法形式的基本語言分析，與是否選取自訂分析無關。在基本文件分析期間，會使用 tt_core_typesystem 說明，並會在共用分析結構中新增下列資訊，您可以在後續的自訂分析中使用此資訊：

- 類型 com.ibm.es.tt.DocumentMetaData 的文件中間資料。
- 文件結構資訊，例如類型 uima.tt.SentenceAnnotation 及 uima.tt.ParagraphAnnotation 的句子及段落註解。
- 詞彙註解，例如類型 uima.tt.TokenAnnotation 的記號及複合項。

tt_core_typesystem 說明適用於大部分文件分析處理作業。

如果想要將集合處理作業變更爲進階分析模式，您可以併入下列兩種類型系統。類型系統主要包括未在基本語言處理作業期間建立的其他特性。

- tt_extension_typesystem，包括更多的記號、詞形、段落及句子特性資訊

- dlt_core_typesystem，包含部分 IBM LanguageWare 延伸註解類型，例如，URL 及位址。它還包括不常使用的形態特性。

tt_core_typesystem

下列類型及特性定義於 tt_core_typesystem 說明中：

uima.tcas.DocumentAnnotation

文件註解包含文件中間資料，並具有下列特性：

- 種類，由文字分類器新增的文件種類。每一個新增的種類都是 com.tt.CategoryConfidencePair 類型
- languageCandidates，剖析期間自動偵測到的文件語言。這些語言會新增至類型 com.tt.LanguageConfidencePair 的清單中，其中最先列示的是最有可能的語言
- id，文件 ID，例如 URL

uima.tt.TTAnnotation

這是在 tt_core_typesystem 中定義之註解的根類型。它的超類型是 uima.tcase.Annotation。有下列類型：

uima.tt.DocStructureAnnotation

文件結構的相關註解。它具有下列次類型：

uima.tt.SentenceAnnotation

句子

uima.tt.ParagraphAnnotation

文件段落

uima.tt.LexicalAnnotation

詞彙註解，例如記號或多字表示式。它具有下列次類型：

uima.tt.TokenLikeAnnotation

單一記號註解可以具有下列特性：

- tokenProperties，記號內容
- lemma，字詞的詞形或詞幹
- normalizedCoveredText，所涵蓋文字的正常化表示法

此註解類型具有下列次類型：

uima.tt.TokenAnnotation

要從複合部分識別的實際記號。

uima.tt.CompPartAnnotation

字詞的複合部分。

uima.tt.CompoundAnnotation

複合記號的註解。複合記號通常跨距多個記號註解。

uima.tt.MultiTokenAnnotation

詞彙註解由多個記號組成。此註解類型具有下列次類型：

uima.tt.StopwordAnnotation

停用字的註解。停用字還可以是多字詞單字。

uima.tt.SynonymAnnotation
具有同義字之字詞的註解。它具有特性 synonyms，會列出找到的字詞同義字。

uima.tt.SpellCorrectionAnnotation
具有拼字更正之字詞的註解。它具有特性 correctionTerms，從最可能的更正開始，以排序的順序列出可能的更正。

uima.tt.MultiWordAnnotation
多字字詞的註解。

uima.CAS.TOP
類型系統的 root。它具有下列次類型：

uima.tt.KeyStringEntry
「字串」資料結構的抽象類型。它包括特性 key，包含字串索引及下列次類型：

uima.tt.Lemma
定義檔詞形項目。

uima.tt.CategoryConfidencePair
已找到之種類的信賴度值。它具有下列特性：

- categoryString，種類的名稱
- categoryConfidence，種類的信賴度值
- mostSpecific，指出此種類是否最適合文件的旗號
- taxonomy，衍生出種類之分類架構的名稱

uima.tt.LanguageConfidencePair
已找到之種類的信賴度值。此類型包括特性 languageConfidence、language 及 languageID。

tt_extension_typesystem

tt_extension_typesystem 包括更多進階處理作業的其他文字分析特性。

uima.tt.TokenLikeAnnotation
此註解類型位於 tt_extension_typesystem 中，具有下列特性：

- lemmaEntries 列出記號的所有可能詞形。這些清單項目是 uima.tt.Lemma 類型
- tokenNumber
- stopwordToken

uima.tt.Lemma
此註解是 uima.tt.KeyStringEntry 類型，具有下列特性：

- isStopword 會在詞形是停用字時爲 true
- isDeterminer 會在詞形是限定詞時爲 true
- partOfSpeech。存在下列詞性號碼說明碼：
  - 0：不明
  - 1：代名詞
  - 2：動詞

- 3：名詞
- 4：形容詞
- 5：副詞
- 6：介詞
- 7：感嘆詞
- 8：連接詞

uima.tt.DocStructureAnnotation

文件結構的相關註解。其具有下列次類型：

uima.tt.SentenceAnnotation

文件句子。其具有特性 `sentenceNumber`。

uima.tt.ParagraphAnnotation

文件段落。其具有特性 `paragraphNumber`。

dlt_extension_typesystem

`dlt_extension_typesystem` 包括 IBM LanguageWare 使用的其他特性。

uima.tt.LexicalAnnotation

此註解具有下列次類型：

uima.tt.TokenLikeAnnotation

在 `dlt_extension_typesystem` 中，此註解具有下列特性：

- `synonymEntries`
- `frost_TokenType`
- `inflectedForms`
- `spellAid`
- `decomposition`

com.ibm.dlt.uimatypes.FilePath

com.ibm.dlt.uimatypes.Email

com.ibm.dlt.uimatypes.Number

com.ibm.dlt.uimatypes.URL

com.ibm.dlt.uimatypes.Date

com.ibm.dlt.uimatypes.Time

com.ibm.dlt.uimatypes.Tel

com.ibm.dlt.uimatypes.Currency

com.ibm.dlt.uimatypes.Acronym

uima.tt.TokenLikeAnnotation

此註解類型位於 `dlt_extension_typesystem` 中，具有下列類型：

com.ibm.dlt.uimatypes.MWU

此類型由 IBM LanguageWare 用來註解多字表示式。

uima.tt.KeyStringEntry

字串註解。其具有下列次類型：

uima.tt.Lemma

它具有下列特性：

- frost_Constraints，限制旗號
- frost_MorphBitMasks，包含形態位元遮罩陣列
- frost_ExtendedPOS，詞性資訊的延伸部分，例如日文的 JPOS 及中文的 CPOS
- frost_JKom，包含日文形態資料
- ，包含日文起始分析資料frost_JPStart，
- morphID，包含詞形屬性

uima.tcas.Annotation

其具有下列次類型：

com.ibm.dlt.uimatypes.Decomp_Analysis

複合項的完整結構分析。其具有下列特性：

- headComponentIndex，複合項的標頭元件
- route，包含組成單一分解路徑的記號清單

相關參考

『類型系統說明範例』

類型系統說明解釋在自訂分析中所使用的特性結構 (代表分析結果的基礎資料結構)。

類型系統說明範例

類型系統說明解釋在自訂分析中所使用的特性結構 (代表分析結果的基礎資料結構)。

類型系統說明必須是分析引擎保存檔 (.pear 檔) 的一部分，該檔案是從 UIMA 環境匯入至企業搜尋。

下列類型系統說明範例說明治安報告，其中包含嫌犯、犯案地點、犯案時間及日期等相關資訊：

相同的範例類型系統說明會用於所有文字分析主題，這些主題討論使用自訂分析時所能選取的不同對映類型。

```
<?xml version="1.0" encoding="UTF-8"?>
<typeSystemDescription>
   <name>Police Reports Type System</name>
   <description>Type system description for
        police reports</description>
   <version>1.0</version>
   <types>
     <typeDescription>
       <name>com.ibm.omnifind.types.PoliceReport</name>
       <description>Annotates a police report</description>
       <supertypeName>uima.tcas.Annotation</supertypeName>
       <features>
         <featureDescription>
           <name>time</name>
           <description>Time the crime was reported to have happened
              </description>
           <rangeTypeName>com.ibm.omnifind.types.Time</rangeTypeName>
         </featureDescription>
         <featureDescription>
           <name>date</name>
           <description>When the crime happened</description>
           <rangeTypeName>com.ibm.omnifind.types.Date</rangeTypeName>
```

```
    </featureDescription>
    <featureDescription>
      <name>location</name>
      <description>Where the crime took place</description>
      <rangeTypeName>com.ibm.omnifind.types.City</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>knownSuspects</name>
      <description>Contains annotations of type Suspect</description>
      <rangeTypeName>uima.cas.FSArray</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>crimeDescription</name>
      <description>Short description of the crime</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.City</name>
  <description>The name of a city</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>cityName</name>
      <description>The name of the city</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>cityDistrict</name>
      <description>The name of the district</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.Person</name>
  <description>A person annotation</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>role</name>
      <description>For example, suspect or witness</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>firstName</name>
      <description>The first name of the person</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>surName</name>
      <description>The surname of the person</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>title</name>
      <description>For example, Mr. or Ms.</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>gender</name>
      <description>Male or female</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
```

```
<typeDescription>
  <name>com.ibm.omnifind.types.Suspect</name>
  <description>A found suspect</description>
  <supertypeName>com.ibm.omnifind.types.Person</supertypeName>
  <features>
    <featureDescription>
      <name>description</name>
      <description>Suspect description,
    for example, bearded with dark glasses</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.Date</name>
  <description>A date</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>year</name>
      <description>The year, for example, 2005</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>month</name>
      <description>The month in digits, for example, 7</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>day</name>
      <description>The day in digits</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>dayOfWeek</name>
      <description>The day of the week, for example, Monday</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>quarter</name>
      <description>The quarter, for example, Q1-2005</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>englDate</name>
      <description>Date as mm/dd/yyyy</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.Time</name>
  <description>A time</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>hours</name>
      <description>Hours from 00-23</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>minutes</name>
      <description>Minutes in the hour</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>timeOfDay</name>
```

```
        <description>Time periods, such as morning, noon</description>
        <rangeTypeName>uima.cas.String</rangeTypeName>
      </featureDescription>
    </features>
  </typeDescription>
 </types>
</typeSystemDescription>
```

分析及搜尋中的 XML 標記

您可以將文件中 XML 結構的資訊直接對映至共用分析結構，而不必撰寫 UIMA 註解程式。

如果集合中的文件是 XML，且您要在文字分析或語意搜尋時利用 XML 標記，則可以選擇下列選項：

原生 XML 搜尋

如果要在語意搜尋期間依照文件中的顯示原樣來使用所有 XML 標示及屬性，則請使用此選項。比方說，如果您的帳單文件包含 <addressee> 元素，則啓用原生 XML 搜尋可讓您在語意搜尋查詢中使用此標示，以在此元素中搜尋特定的客戶名稱。

使用此選項，則會在共用分析結構中使用 com.ibm.es.tt.MarkupTag 類型來顯示文件的 XML 結構。針對每一個 XML 標示，都會建立此類型的註解。這個註解含有標示名稱、其屬性及屬性內容。此資訊一定會加以索引，且可以存取它來執行語意搜尋。

原生 XML 搜尋不需要對映配置檔。您可以從企業搜尋的管理主控台啓用原生 XML 搜尋。

XML 元素到共用分析結構的對映

在下列情況下，請使用此選項：

- 某些 XML 元素的語意明確，且可以用於進一步的文字分析步驟。這些分析步驟可以直接在從 XML 結構所建立的註解和特性上運作，且會和潛在的不同原始文件格式隔離。例如，帳單文件中的元素 <addressee> 通常含有客戶名稱。使用 XML 元素到共用分析結構的對映，可以將此元素的內容直接對映至類型 Customer 的註解。然後，註解程式可以利用 Customer 註解周遭的資訊，推斷「客戶所在地」關係。
- 您想要將客戶註解程式的處理作業範圍限制在 XML 輸入中的指定區域。例如，您可能要將 <technicianComment> 標示的內容分析僅限制在偵測汽車問題的註解程式。
- 您想要將文字分析處理程序和後續的搜尋限制於 XML 文件的特定部分，然後過濾出無關或非文字內容。
- 您想要將含有不同名稱的 XML 標示對映至要在語意搜尋中使用的一般跨距。例如，將 <mainHeading> 或 <doc> 對映至標題。

在這些情況下，您必須建立 XML 元素到共用分析結構的對映檔，以定義哪個 XML 元素對映哪個特性結構。您在對映檔中定義的特性結構會在剖析文件時建立，並由自訂註解程式存取。

您可以將多個 XML 元素到共用分析結構的對映檔用於文件集合。哪個對映檔用於哪個 XML 文件，是由 <identifier> 元素決定的。對映檔中的 <identifier> 元素必須符合 XML 文件中的根元素。例如，如果文件的根元素是 doc，則對映檔中 <identifier> 元素的值也必須是 ″doc″。

如果找不到相符的項目，則程式會搜尋 <identifier> 元素設爲 Default 的對映檔。如果找不到預設的對映，則文件的文字區段 (沒有標示資訊) 會對映至共用分析結構中的文件註解。

如果要擷取只在文件相關部分才有的資訊，並忽略無關的部分，只需指定文件中哪些 XML 元素包含相關資訊即可。這稱爲內容擷取。例如，您可以擷取在標題和主體元素中指定的輸入，並忽略作者、日期、ID 和發佈者中的輸入。

內容擷取可以增進下列 XML 文件類型的分析處理程序：

- 包含大量內容且不適用於分析的文件，例如，二進位附件。使用內容擷取會大幅降低文件大小，加速處理程序並避免因不當資料而產生的分析錯誤。
- 文件文字中散佈著無關文字的文件，例如，在 <note> 標示中包含編輯資訊的文件。分析文件內容時，忽略此資訊可以有更好的結果。

使用原生 XML 搜尋與 XML 元素到共用分析結構之對映中的內容擷取選項彼此矛盾，因爲要考慮所有內容或只能考慮指定的內容。如果指定內容擷取，則會忽略原生 XML 對映。沒有內容擷取，您可以同時使用 XML 元素到共用分析結構的對映與原生 XML 搜尋。

您在配置檔中使用的所有類型和特性，都必須在自訂分析步驟的類型系統說明中加以說明。您可以使用「元件描述子編輯器 Eclipse」外掛程式，在 UIMA 環境中建立類型系統描述子。此外掛程式可讓您無需瞭解必要的 XML 語法，便可以建立描述子檔案。

建置並測試自訂分析之後，請使用 UIMA PEAR (「處理程序引擎保存檔」) 產生精靈，建立包含自訂分析檔案 (包括類型系統說明) 的保存檔。然後，您可以使用企業搜尋的管理主控台，將自訂分析保存檔及 XML 元素到共用分析結構的對映檔上載至企業搜尋。

相關工作

『建立 XML 元素到共用分析結構的對映檔』
在 XML 到共用分析結構的對映檔中，您可以使用所有配置選項，將 XML 對映至 UIMA 資料類型。

建立 XML 元素到共用分析結構的對映檔

在 XML 到共用分析結構的對映檔中，您可以使用所有配置選項，將 XML 對映至 UIMA 資料類型。

關於本作業

下列範例中會顯示 XML 到共用分析結構的對映檔。

治安報告範例含有用於犯罪類型、犯罪日期、犯罪地點、報告員警、該員警任職的警方轄區、嫌犯說明及摘要的 XML 標示。後面接著主體區段。例如：

```
<report>
 <doc>
<crimeType>Car theft</crimeType>
<crimeDate>04/23/05 09:23 pm</crimeDate>
<crimeLocation>27 Main Street, Brynston, Springfield, New Jersey</crimeLocation>
```

```
<reportingOfficer rank="Lt">Jakob
<lastName>Collins</lastName>
  </reportingOfficer>
<policePrecinct>14th Precinct</policePrecinct>
<suspectDescription>Male, dark haired, dark glasses,
    blue jeans with dark, probably black,
    jacket</suspectDescription>
<abstract>A Mercedes CLK was stolen on 04/23/2005 from a parking
   lot in front of the Blue Lagoon restaurant on
27 Main Street, Brynston.(serial number: 32 2761 50871)</abstract>
<body>A Mercedes CLK was stolen on 04/23/2004 from a parking
   lot in front of the Blue Lagoon restaurant on 27 Main Street,
   Brynston.(serial number: 32 2761 50871)

It has a black color and wide Michelin tires.

Eyewitnesses in front of the restaurant saw two darkly dressed
males drive away in the car at high speed. The car was
found abandoned on Aliway Ave in Brooklyn. The fuel tank was empty.
The seats were badly stained and the back seat was vandalized.
Nothing was stolen out of the car....</body>
 </doc>
 <image>
  <--! image of the crime scene as a base64-encoded string -->
 </image>
</report>
```

根據範例報告，XML 到共用分析結構的對映檔可能具有下列結構。範例使用爲治安報告實務範例所定義的類型系統。

```
<?xml version="1.0"?>
<xmlCasInitializerConfiguration
xmlns="http://www.ibm.com/2005/uima/jedii_ci_xml">

<identifier>Default</identifier>
<description>Sample configuration</description>

 <contentElements>
<element>/report/doc</element>
 </contentElements>

 <elementToTypeMappings>
   <elementToTypeMapping>
<element>//doc//reportingOfficer</element>
<type>com.ibm.omnifind.types.Person</type>
     <featureValueAssignment>
<feature>role</feature>
<basicValue default="Reporting officer">
           </basicValue>
     </featureValueAssignment>
     <featureValueAssignment>
<feature>gender</feature>
       <basicValue default="male"
useAttributeValue="sex"/>
     </featureValueAssignment>
     <featureValueAssignment>
<feature>surName</feature>
<values concatenate="true" delimiter=" ">
         <basicValue useAttributeValue="rank"
default="Lt"/>
<basicValue useElementContent="lastName"/>
       </values>
     </featureValueAssignment>
   </elementToTypeMapping>
   <elementToTypeMapping>
<element>//doc</element>
<type>com.ibm.omnifind.types.PoliceReport</type>
```

```
     <featureValueAssignment>
<feature>crimeDescription</feature>
       <basicValue useElementContent="abstract"
trim="true">
           </basicValue>
     </featureValueAssignment>
   </elementToTypeMapping>
 </elementToTypeMappings>

</xmlCasInitializerConfiguration>
```

限制

對映檔分成兩個區段：

<contentElements> 元素

如果要擷取特定內容，請使用此元素。範例對映檔會擷取文件中 <doc> 區段的內容，而忽略文件中的其他區段。在 XML 治安報告中，影像可能會很大，且對文字處理不是非常有幫助。藉由指定 <doc> 作爲內容元素而不指定 <image> 後，會在開始任何文字處理作業之前先過濾出影像。

<elementToTypeMappings>

使用此元素可以指定文件中的哪些個別 XML 元素 (指定於 <elementToTypeMapping> 元素) 要對映至共用分析結構中的哪些特性結構。

如果您使用內容擷取選項，則 <elementToTypeMappings> 區段中指定的 XML 元素必須內含於 <contentElements> 區段中指定的 XML 元素中。

程序

若要建立 XML 到共用分析結構的對映檔，請：

1. 建立 XML 檔。 若要避免 XML 語法錯誤，請使用 XML 編輯器或 XML 編寫工具以驗證 XML。對映檔的 XSD 綱目稱爲 XMLCasInitSchema.xsd，且內含於企業搜尋安裝的 *ES_INSTALL_ROOT*/packages/uima/configuration_xsd/ 中。
2. 在 <xmlCasInitializerConfiguration xmlns="http://www.ibm.com/2005/uima/jedii_ci_xml"> 元素中併入對映。名稱空間 (在 xmlns 屬性中指定) 必須如所示範例一模一樣。
3. 如果您要從文件的區段中擷取特定內容，請新增 <contentElements> 元素及 <elementToTypeMappings> 元素，後者指定要將文件中哪些個別的 XML 元素對映至共用分析區域的哪些特性結構。
4. 新增 <identifier> 元素及 <description> 元素。 ID 會決定要將哪個對映用於哪個 XML 文件。ID 必須含有文件的根元素，如 doc。如果 ID 設爲「預設值」，則文件的根元素就不適用，且對映會套用至任何 XML 文件。
5. 如果要擷取只在文件相關部分才有的資訊，請新增 <contentElements> 元素。它含有下列元件元素：
   - 一或多個 <element> 元素，其中包含文件中 XML 元素的路徑並遵循 XPath 語法，例如 <element>/doc/crimeType</element>。
6. 如果要指定文件中的哪些 XML 元素要對映至共用分析結構中的哪些特性結構，請新增 <elementToTypeMappings> 元素。有下列元件元素：
   - 一或多個 <elementToTypeMapping> 元素。這個元素必須含有下列巢狀元素：
     – <element> 元素，用來指定 XML 元素的路徑並遵循 XPath 語法：前導正斜線 (/) 表示已提供完整路徑。例如，根元素 doc 下的 abstract。兩個正斜線

(//) 表示任何路徑子集。例如，birthDate 必須發生在 reportingOfficer 內，雖然其他元素可以發生在這兩者之間。

- <type> 元素，指定類型系統說明中定義的類型。它必須屬於類型 Annotation。
- 零或多個 <featureValueAssignment> 元素。

7. 在 <featureValueAssignment> 元素中，命名 <feature> 元素中類型爲 String 的特性，並在 <basicValue> 元素中指定值。可以在 <values> 元素之間新增多個 <basicValue> 元素。

   <basicValue> 元素可以具有屬性。這些包括 useAttributeValue、useElementContent、default 及 trim。

如果要使用屬性值作爲特性值，請使用 useAttributeValue。下面示範

```
  <elementToTypeMapping>
<element>/doc//reportingOfficer</element>
<type>com.ibm.omnifind.types.Person</type>
    <featureValueAssignment>
<feature>role</feature>
<basicValue default="Reporting officer"/>
    </featureValueAssignment>
    <featureValueAssignment>
<feature>gender</feature>
      <basicValue default="male" useAttributeValue="sex"/>
    </featureValueAssignment>
  </elementToTypeMapping>
```

產生下列輸出：

- 針對文件中 <doc> XML 標示某處所發生的每一個 <reportingOfficer> XML 標示，都會建立一個 com.ibm.omnifind.types.Person 類型的特性結構。
- 如果 <reportingOfficer> 標示包含屬性 sex，則新建特性結構的特性 gender 會設爲該屬性的值。

請使用屬性 useElementContent 來新增內容作爲特性的值。例如，在下列對映片段中：

```
  <elementToTypeMapping>
<element>//doc</element>
<type>com.ibm.omnifind.types.PoliceReport</type>
    <featureValueAssignment>
<feature>crimeDescription</feature>
      <basicValue useElementContent="abstract" trim="true"/>
    </featureValueAssignment>
  </elementToTypeMapping>
```

<doc> 中元素 <abstract> 所涵蓋的文字會變成特性結構 crimeDescription 的值。將會移除所有前導及尾端空白。

在下列情況下，可以在 <values> 元素之間指定多個值：

- 要設定的特性屬於 StringArray 類型。
- 利用區隔字元屬性，將許多字串連結成一個字串，並因而對映至類型 String 的特性。例如，職稱 Mr. 是常數、名字是屬性值，而 XML 元素會涵蓋姓氏：

```
   <elementToTypeMapping>
  <element>//doc//reportingOfficer</element>
  <type>com.ibm.omnifind.types.Person</type>
      <featureValueAssignment>
  <feature>surName</feature>
        <values concatenate="true" delimiter=" ">
          <basicValue default="Mr."/>
```

```
        <basicValue useAttributeValue="rank"
  default="Lt."/>
  <basicValue useElementContent="lastName"/>
        </values>
      </featureValueAssignment>
    </elementToTypeMapping>
```

字串特性值會如現狀從對映檔中擷取出來。值會保留所有前導或尾端空白。但會裁去類型及特性名稱中的空白。例如，<type>**com.ibm.omnifind.types.Person**</type> 會成爲 <type>**com.ibm.omnifind.types.Person**</type>。

使用 <condition> 元素設定屬性的條件。例如，只有在屬性 armed 設爲 yes 的文件中發生 <suspectDescription> 時，才會建立類型爲 com.ibm.omnifind.types.Person 的特性結構：

```
  <elementToTypeMapping>
<element>//suspectDescription</element>
<type>com.ibm.omnifind.types.Person</type>
    <condition attribute="armed"  value="yes"/>
  </elementToTypeMapping>
```

根據範例治安報告及已定義的對映檔，建立下列特性結構：

com.ibm.omnifind.types.PoliceReport

- covered text: ″Car theft 04/23/05 09:23 pm 27 Main Street, Brynston, Springfield, New Jersey Jakob Collins 14th Precinct Male, dark haired, dark glasses, blue jeans with dark, probably black, jacket A Mercedes CLK was ... Nothing was stolen out of the car.
- begin = 2
- end = 904
- knownSuspects = null
- crimeDescription = ″A Mercedes CLK was stolen on 04/23/2005 from a parking lot in front of the Blue Lagoon restaurant on 27 Main Street, Brynston.(serial number: 32 2761 50871)″

com.ibm.omnifind.types.Person

- covered text = ″Jakob Collins″
- begin = 112
- end = 127
- role = ″Reporting officer″
- firstName = null
- surName = ″Lt Collins″
- gender = ″male″

建立對映檔之後，您必須將它上載至企業搜尋，然後利用企業搜尋管理主控台，來選取含有其他自訂分析選擇的 XML 到共用分析結構對映檔。

相關概念

第 21 頁的『分析及搜尋中的 XML 標記』
您可以將文件中 XML 結構的資訊直接對映至共用分析結構，而不必撰寫 UIMA 註解程式。

相關參考

第 18 頁的『類型系統說明範例』
類型系統說明解釋在自訂分析中所使用的特性結構 (代表分析結果的基礎資料結構)。

文字分析結果

所有文字分析結果都儲存在共用分析結構中。

註解程式通常會在共用分析結構中讀取和寫入。共用分析結構消費者 (*CAS 消費者*) 僅從共用分析結構進行讀取。CAS 消費者會對儲存在共用分析結構的分析結果執行最終的處理作業。企業搜尋含有兩種 CAS 消費者：

- 在搜尋引擎中索引共用分析結構內容的消費者。此消費者需要共用分析結構到索引的對映檔，您可以在企業搜尋管理主控台上使用自訂文字分析來選取該檔案。
- 在關聯式資料庫中輸入特定分析結果的消費者。此消費者也需要共用分析結構到資料庫的對映檔，您可以在企業搜尋管理主控台上使用自訂文字分析選項選取該檔案。

必要時，您可以在企業搜尋中部署自訂 CAS 消費者。如需如何撰寫消費者的相關資訊，請參閱 UIMA 文件。若要瞭解如何在企業搜尋中上傳及使用消費者，請參閱 IBM UIMA developerWorks 網站，網址爲 http://www.ibm.com/developerworks/db2/zones/db2ii/。

相關概念

第 31 頁的『自訂分析結果的索引對映』
對文件集合執行自訂分析後， 您可以使用企業搜尋中的搜尋引擎，從儲存在共用分析結構 (以自訂分析演算法建立) 中的資訊建置索引。

第 37 頁的『所選分析結果的資料庫對映』
在企業搜尋中分析文件之後，您可以將選取的文字分析結果儲存在具有 JDBC 功能的資料庫中。

特性路徑

特性路徑可讓您存取共用分析結構中的特性值，類似用來存取 XML 文件中 XML 元素的 XPath 陳述式。

如果您要存取結合複式特性 (例如，陣列值或指向另一個特性結構的特性) 的特性結構時，特性路徑是很有用的。利用特性路徑，您可以直接關聯特性值與特性結構，並將此值儲存在語意搜尋索引或資料庫中。

例如，考慮識別汽車及其樣式的註解程式。它會建立類型 `car` 且含有屬性 `make` 的註解。然而，`make` 不含實際公司 (例如， Chevrolet)，但含有類型 `Company` 的特性結構，而此特性結構本身含有字串值的屬性 `companyname`。若要啓用結合汽車名稱及公司名稱的語意查詢，可以使用特性路徑 `make/companyname` 將 `companyname` 的值連接至汽車註解所產生的汽車跨距。使用 '/car[@make="Chevrolet"]' 可以啓用下列查詢「給我含有 Chevrolet 製造的 cars 文件」。

特性路徑是一連串具有下列內容的特性名稱 (f1/.../fn)：

- 特性路徑的值可以是 String、Integer、Float 或其中一種類型的陣列。
- 路徑中的所有特性 (從 f1 到 fn-1) 都必須具有複數類型，亦即，屬於類型 `uima.cas.TOP`，`uima.cas.FSArray`，`uima.cas.FSList` 或其中一個次類型。
- 路徑中的最後一個特性可以包含複數類型，此外，它也可以包含 `uima.cas.Float`、`uima.cas.Integer`、 `uima.cas.String`、`uima.cas.FloatArray`、

uima.cas.IntegerArray、uima.cas.StringArray、uima.cas.FloatList、uima.cas.IntegerList 或 uima.cas.StringList 的 (次) 類型。

- 您可以選擇性地鍵入特性。完整的類型名稱必須附加到特性名稱的前面，且必須以冒號區隔。例如，f1/com.ibm.es.SomeType:f2/.../fn。

您可以縮小特定特性的類型範圍。例如，考慮類型 uima.cas.TOP 的特性 additionalInfo。如果您知道特性 additionalInfo 的值實際上是屬於類型 EmployeeInfo，其中含有特性 salary，則可以利用 additionalInfo/EmployeeInfo:salary 來存取此特性。請注意，在本例中特性路徑 additionalInfo/salary 會造成錯誤，因爲 salary 尚未在類型 uima.cas.TOP 中定義。

陣列或清單值的特性具有下列額外內容：

- 使用方括弧 ([<number>]) 來選取陣列或清單中的特定元素。陣列從零 (0) 開始。例如，若要選取公司 (companies) 陣列中的第一個元素，請使用 companies[0]。您可以使用特殊標記 [last] 來選取陣列中的最後一個項目，與它的大小無關，例如 companies[last]。
- 使用空的方括弧 ([]) 來表示所有元素。在一個特性路徑中，只容許使用一個空的方括弧 ([])。比方說，如果有嫌犯陣列，則特性路徑 knownSuspects[]/com.ibm.omnifind.types.Suspect:surName 會將所有嫌犯的姓氏收集到 String 陣列。
- 在檢索期間使用傳回陣列的特性路徑時，則會連結 (以空格區隔) 陣列元素並寫入索引作爲單一的多字詞屬性或欄位。
- 必須鍵入特性路徑中的下一個元素。類型名稱是陣列中的元素類型。例如，考慮類型 Info 的特性結構。此類型含有名稱爲 companies 的特性，其範圍是 FSArray。陣列的元素屬於類型 Company。換言之，Company 具有名稱爲 profit 的特性。若要取得第三家公司的利潤，請寫入 (利用完整的類型名稱) companies[2]/Company:profit。

內建特性

內建特性是預先定義的特性名稱，這些名稱具有特殊的語意。它們可以用來存取特性結構本身沒有的資訊，例如，特性結構的類型或註解的涵蓋文字。它們可以當成特性路徑中的最後一個或唯一的元素使用。

下列內建特性可以在兩個對映配置檔中使用：

- fsId() 傳回特性結構的 ID。傳回的 ID 是整數 (32 位元)。請使用這個內建特性來存取文件中完全符合查詢的部分。
- typeName() 以字串傳回共用分析結構物件類型。類型是包含任何名稱空間字首的完整類型名稱，例如 uima.tcas.Annotation。在資料庫環境定義中，如果您在相同的直欄中儲存類型及次類型，且想要知道註解或特性結構的實際類型，則 typeName() 特別有用。下列範例在角色直欄中儲存了人員類型，如*嫌犯*或*證人*。

```
<explicitMappingRule applyToSubTypes="false">
<type>com.ibm.omnifind.types.Person</type>
<table>sample.person</table>
   <featureMappings>
     <featureMapping>
<feature>typeName()</feature>
<column>role</column>
     </featureMapping>
   </featureMappings>
 </explicitMappingRule>
```

- coveredText() 傳回共用分析物件所橫跨的文字。 coveredText() 只適用於註解及其次類型。請勿在註解類型未納入的特性結構上使用這個內建特性。下列範例在 suspectName 直欄中儲存了嫌犯名稱。

```
<implicitMappingRule applyToSubTypes="false">
<type>com.ibm.omnifind.types.Suspect</type>
<relation>sample.person</relation>
   <featureMappings>
     <featureMapping>
<feature>coveredText()</feature>
<column>suspectName</column>
<length>128</length>
     </featureMapping>
   </featureMappings>
 </implicitMappingRule>
```

- [] 將控點傳回給現行儲存區項目 (陣列或清單)。特性暗示疊代，這表示資料庫表格中有項目，或陣列或清單的每一個元素有索引。下列範例取自資料庫對映檔的共用分析結構，該檔案中也容許內建函數 [:index]。

```
<implicitMappingRule applyToSubTypes="false">
<type>uima.cas.FSArray</type>
<table>sample.knownSuspects</table>
   <featureMappings>
     <featureMapping>
<feature>uniqueId()</feature>
<column>arrayId</column>
     </featureMapping>
     <featureMapping>
<feature>[:index]</feature>
<column>arrayIndex</column>
     </featureMapping>
     <featureMapping>
<feature>[]/com.ibm.omnifind.types.Suspect:uniqueId()</feature>
<column>suspectId</column>
     </featureMapping>
   </featureMappings>
 </implicitMappingRule>
```

下列內建特性只能用於共用分析結構到資料庫的對映檔：

- uniqueId() 傳回特性結構的廣域唯一 ID。傳回的唯一 ID 是固定長度的字串 (27 個字元)，且是 fsId()、docId()、docTimestamp() 及現行片段號碼的連結，因爲文件可以在企業搜尋中分成多個共用分析結構。

  傳回的字串可以包含 ″a-z″ 及 ″A-Z″ 之間的任何字元、數字 ″0-9″、分號 (″;″) 及冒號 (″:″)。

  uniqueId() 的結果可以當成表格的主要索引鍵使用。

- objectId() 傳回註解或特性結構的 ID。 objectId() 類似 uniqueId()，只是它不含 docTimestamp() 的結果。傳回的 ID 只在文件剖析過一次的集合中是唯一的。如果需要在所有文件及文件版本中都是唯一，則必須使用 uniqueId()。

  內建特性 objectId() 的傳回字串是 16 個字元的固定長度，且可以含有 a-z 及 A-Z 之間的任何字元、數字 0-9、分號 (″;″) 及冒號 (″:″)。

  如果 uniqueId() 或 objectId() 參照的特性結構是空的，則會採用定義於資料庫表格定義中的預設值，而不會儲存被參照類型的空物件。

- docId() 傳回文件 ID。傳回的值是屬於整數類型 (32 位元)。

下列範例顯示這些內建特性：

```
<explicitMappingRule applyToSubTypes="true">
<type>com.ibm.omnifind.types.PoliceReport</type>
<table>sample.PoliceReport</table>
   <featureMappings>
     <featureMapping>
<feature>uniqueId()</feature>
<column>policeReportId</column>
     </featureMapping>
     <featureMapping>
<feature>docId()</feature>
<column>policeReportDocId</column>
     </featureMapping>
   </featureMappings>
 </explicitMappingRule>
```

- `docUri()` 傳回文件 URI。
- `docTimestamp()` 傳回處理文件時的時間 (毫秒)。若要追蹤文件版本 (例如，如果您要知道使用的文件版本是否爲搜索器所傳送的最新版本)，這個內建特性是很有用的。

```
<explicitMappingRule applyToSubTypes="false">
<type>com.ibm.omnifind.types.PoliceReport</type>
<relation>sample.PoliceReport</relation>
   <featureMappings>
     <featureMapping>
<feature>uniqueId()</feature>
<column>policeReportId</column>
     </featureMapping>
     <featureMapping>
<feature>docTimestamp()</feature>
<column>reportVersion</column>
     </featureMapping>
   </featureMappings>
 </explicitMappingRule>
```

- `parentId()` 傳回含有儲存區對映的特性結構之 `fsId()`。 `parentId()` 只有在儲存區對映的環境定義中有效。
- `uniqueParentId()` 傳回儲存區對映所含的註解或特性結構的 `uniqueId()`。這個內建特性也只適用於儲存區對映的環境定義中。
- `[:index]` 傳回現行儲存區項目 (陣列或清單) 的索引。

相關工作

第 47 頁的『擷取文件中符合語意搜尋查詢的部分』
藉由將相關的特性結構同時對映至索引與資料庫，並在語意搜尋查詢中指定跨距，您可以只擷取文件中完全符合查詢的部分。

過濾器

過濾器可用來限制共用分析結構到索引的對映檔，以及共用分析結構到資料庫的對映檔中的對映規則。只有在過濾條件爲眞時，分析結果才會加入索引或 JDBC 表格。

`<filter>` 元素是選用的，用來只將對映限制於有特定屬性值的特性。如果要將屬性當成索引或新增至資料庫的開關時，這是非常有用的。例如，可在類型 `EntityAnnotation` 的註解中記錄人員及組織。設定稱爲類型的特性爲人員或組織。若只要擷取人員而不要組織，您可以將下列過濾條件加入對映規則：

```
<filter syntax="FeatureValue">type = "person"</filter>
```

每一個過濾表示式的格式如下：

```
<FeaturePath> <Operator> <Literal>
```

其中：

- FeaturePath 是共用分析結構中的特性路徑
- 運算子是 =、!=、<、<=、> 或 >=。請注意，< (且只有 <) 必須表示爲 <。
- 文字是整數、浮點數 (不支援指數語法) 或以雙引號括住的字串文字，內含引號及使用反斜線作爲跳出字元的反斜線。

<FeaturePath>、<Operator> 及 <Literal> 必須以空格區隔。

下列範例是有效的過濾器：

- `<filter syntax="FeatureValue"> foo = "hello world" </filter>`

  特性 foo 含有字串 hello world。
- `<filter syntax="FeatureValue"> foo < 42 </filter>`

  特性 foo 具有小於 42 的整數值。
- `<filter syntax="FeatureValue"> make/company = "Chevrolet" </filter>`

  特性路徑 make/company，其中特性 make 所含的特性結構中具有隨附值 Chevrolet 的特性。
- `<filter syntax="FeatureValue"> bar7 >= 0.5 </filter>`

  特性 bar7 具有大於或等於 0.5 的浮點值。

自訂分析結果的索引對映

對文件集合執行自訂分析後，您可以使用企業搜尋中的搜尋引擎，從儲存在共用分析結構 (以自訂分析演算法建立) 中的資訊建置索引。

將分析結果對映至企業搜尋索引中的欄位、文字跨距及屬性，可讓您在查詢中使用此資訊。合併自訂分析與可以索引單字和文字跨距的企業搜尋，可啓用語意搜尋。

使用共用分析結構到索引的對映檔，您可以決定要索引共用分析結構中的哪些分析結果。

您可以使用不同的樣式，將共用分析結構中的特性結構對映至企業搜尋索引。

註解 如果利用註解樣式來索引共用分析結構中的特性結構，則所有指定類型的註解都會在索引中儲存爲可搜尋的跨距。

比方說，如果跨距特定文字區域的特性結構屬於 person 類型，並使用註解樣式加以檢索，則下列查詢是可行的：

表 2. 範例查詢

必要資訊	可能的查詢
給我至少含有一個人員名稱的所有文件	<person/>
給我人員註解中含有上司的所有文件	<person>boss</person>
給我在我的競爭對手之一的相同句子中提到 Lang 的所有文件	<sentence><person>Lang</person> <competitor/></sentence>

特性結構的屬性也可以當成跨距的一部分來索引。例如，考慮偵測汽車的註解程式，並將汽車樣式儲存爲 car 註解的特性 make。這會啓用下列類型的查詢：「給我提到 Chevrolet 汽車樣式的文件」。

欄位 如果要在搜尋期間，利用企業搜尋的欄位搜尋功能來存取特性結構的內容，請使用此樣式。在此方式中，特性結構的內容可以顯示在搜尋結果中，或用於參數搜尋。

比方說，如果將藥品劑量對映至參數欄位，則可以使用下列查詢：「給我談到某些藥品劑量超過 100 毫克的所有文件」。

中斷 如果要將特定的特性結構解譯爲清除區隔字元 (例如，小節或段落)，則請使用此樣式。企業搜尋預設爲偵測句子及段落。只有在您的自訂分析在文件中偵測到額外的結構元素且您必須有不同的解譯方式時，才能使用此樣式。

也可以使用分析結果來影響企業搜尋中的文件相關性排序，即使是在簡式關鍵字查詢中。此程序分成兩步驟：

1. 使用「註解」或「欄位」對映樣式，將特性結構對映至可搜尋的跨距或欄位。
2. 使用企業搜尋管理主控台來定義 boost 類別，並將跨距或欄位名稱對映至這個 boost 類別。

如果使用者輸入的搜尋字詞內含於特性結構中，則文件的排序會較高。例如，考慮偵測人員和公司名稱的註解程式。將這些特性結構對映至跨距 (如「人員」及「公司」)，然後將這些跨距對映至 boost 類別，則 ″gap″ 的搜尋結果會讓談到 ″Gap″ 公司的文件排序高於只含有 ″gap″ 一詞的文件。

撰寫共用分析結構到索引的對映檔之後，您便可以使用管理主控台將它上載至企業搜尋。

相關工作

『建立共用分析結構到索引的對映檔』
使用共用分析結構到索引的對映檔，您可以決定要索引共用分析結構中的哪些分析結果以啓用搜尋。

建立共用分析結構到索引的對映檔

使用共用分析結構到索引的對映檔，您可以決定要索引共用分析結構中的哪些分析結果以啓用搜尋。

關於本作業

共用分析結構到索引的對映檔是 XML 格式。範例共用分析結構到索引的對映檔是以針對治安報告實務範例定義的類型系統爲基礎。

```
<?xml version="1.0" encoding="UTF-8"?>
<indexBuildSpecification
xmlns="http://www.ibm.com/of/822/consumer/index/xml">
 <skipCondition>
<type>com.ibm.uima.tt.DocumentAnnotation</type>
<filter syntax="FeatureValue">toBeprocessed = 0</filter>
 </skipCondition>

 <indexBuildItem>
   <name>com.ibm.omnifind.types.Person</name>
   <indexRule>
<style name="Annotation">
       <attributemappings>
```

```
         <mapping>
<feature>role</feature>
<indexName>role</indexName>
         </mapping>
         <mapping>
<feature>title</feature>
<indexName>title</indexName>
         </mapping>
         <mapping>
<feature>gender</feature>
<indexName>gender</indexName>
         </mapping>
       </attributemappings>
     </style>
   </indexRule>
 </indexBuildItem>
 <indexBuildItem>
   <name>com.ibm.omnifind.types.Suspect</name>
   <indexRule>
<style name="Annotation"/>
<style name="Field">
       <attribute name="parametric" value="false"/>
       <attribute name="fieldSearchable"
value="true"/>
       <attribute name="returnable" value="true"/>
     </style>
   </indexRule>
 </indexBuildItem>
 <indexBuildItem>
   <name>com.ibm.omnifind.types.City</name>
   <indexRule>
<style name="Annotation">
       <attributemappings>
         <mapping>
<feature>cityDistrict</feature>
<indexName>district</indexName>
         </mapping>
       </attributemappings>
     </style>
   </indexRule>
 </indexBuildItem>
 <indexBuildItem>
<name>com.ibm.omnifind.types.Date</name>
   <indexRule>
<style name="Field">
       <attribute name="fixedName" value="Date"/>
       <attribute name="fieldSearchable"
value="true"/>
       <attribute name="returnable" value="true"/>
     </style>
<style name="Field">
       <attribute name="fixedName" value="hour"/>
       <attribute name="valueFeature"  value="hour"/>
       <attribute name="parametric" value="true"/>
     </style>
   </indexRule>
<filter syntax="FeatureValue">year="2005"</filter>
 </indexBuildItem>
 <indexBuildItem>
   <name>com.ibm.omnifind.types.PoliceReport</name>
   <indexRule>
<style name="Annotation">
       <attribute name="fixedName"
value="PoliceReport"/>
       <attributemappings>
         <mapping>
<feature>crimeDescription</feature>
```

```
<indexName>crimeDescription</indexName>
         </mapping>
         <mapping>
<feature>time/coveredText()</feature>
<indexName>time</indexName>
         </mapping>
         <mapping>
<feature>date/englDate</feature>
<indexName>date</indexName>
         </mapping>
         <mapping>
<feature>location/coveredText()</feature>
<indexName>location</indexName>
         </mapping>
         <mapping>
<feature>knownSuspects[]/com.ibm.omnifind.types.Suspect:surName</feature>
<indexName>suspectsLastNames</indexName>
         </mapping>
       </attributemappings>
     </style>
   </indexRule>
 </indexBuildItem>
</indexBuildSpecification>
```

限制

共用分析結構到索引的對映檔必須包含您能夠在查詢中搜尋的所有分析結果。

程序

若要建立共用分析結構到索引的對映檔，請：

1. 建立 XML 檔。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。對映檔的 XSD 綱目稱爲 CasToIndexMapping.xsd，且內含於企業搜尋安裝的 *ES_INSTALL_ROOT*/packages/uima/configuration_xsd/ 中。
2. 在 `<indexBuildSpecification xmlns="http://www.ibm.com/of/822/consumer/index/xml">` 元素中併入對映。名稱空間 (在 xmlns 屬性中指定) 必須如所示範例一模一樣。
3. 新增 `<skipCondition>` 元素，以根據特定特性值禁止索引某些文件。這是選用的元素。在範例中，將不會檢索資料結構類型爲 `com.ibm.uima.tt.DocumentAnnotation` 且特性 `toBeProcessed` 設爲 0 的文件。
4. 新增一或多個 `<indexBuildItem>` 元素，其中包含共用分析結構中某一特定特性結構對索引中某一結構的對映。
5. 儲存並驗證 XML 檔。

`<indexBuildItem>` 元素

共用分析結構到索引的對映檔包含一或多個 `<indexBuildItem>` 元素。每一個說明均說明共用分析結構中特定特性結構對索引 (跨距或欄位) 中某一結構的對映。

`<name>` 元素包含特性結構類型。指定類型的方式有兩種：

- 完整類型名稱。例如，`com.ibm.omnifind.types.Suspect`
- 萬用字元。例如，`com.ibm.omnifind.types.*`。萬用字元只能在類型規格的尾端加入。

只使用 uima.tcas.Annotation 的次類型作爲索引建置項目。如果特性結構是次類型 uima.cas.TOP (而不是屬於 uima.tcas.Annotation)，您可以利用從註解開始的特性路徑來存取這個特性結構。

如果類型 A 是類型 B 的次類型 (在範例中，com.ibm.omnifind.types.Suspect 是 com.ibm.omnifind.types.Person 的次類型)，且兩種類型均定義了 <indexBuildItem> 元素 Ia 及 Ib，則處理作業會如下所示：

- 定義於 Ib 的每一個索引規則會套用於類型 B 的特性結構和類型 A 的特性結構
- 定義於 Ia 的每一個索引規則只會套用於類型 A 的特性結構

在範例中，針對 com.ibm.omnifind.types.Person 註解所定義的 <indexBuildItem> 元素也會套用至 com.ibm.omnifind.types.Suspect 註解。建立嫌犯註解的兩個跨距：一個命名爲「人員」，另一個則爲「嫌犯」。

<filter> 元素是選用的，用來只將 <indexBuildItem> 對映限制於具有特定屬性值的特性結構。如果要將屬性當成索引的開關時，這是非常有用的。例如，可在類型 EntityAnnotation 的註解中記錄人員及組織。設定稱爲**類型**的特性爲**人員**或**組織**。若只要擷取人員而不要組織，您可以新增下列過濾條件：

```
<filter syntax="FeatureValue">type = "person"</filter>
```

此外，您還可以選擇檢索不同跨距名稱下的人員和組織，例如：person 及 organization。若要執行此作業，請定義類型爲 EntityAnnotation 的兩個 <indexBuildItem> 元素，然後在 type 特性上使用兩個過濾器以觸發人員或組織。

<indexRule> 元素

每一個 <indexBuildItem> 元素都包含一個 <indexRule> 元素。每一個 <indexRule> 元素都包含將共用分析結構中的特性結構對映至索引作爲欄位、註解及中斷樣式所需的所有資訊。註解及欄位樣式支援許多屬性。但您不能在企業搜尋中使用「UIMA 軟體開發套件」支援的詞彙樣式。

若爲註解及欄位樣式，當您在索引中指定註解或欄位名稱時，可以使用下列選擇方案：

- 如果要在索引中以同一個名稱存取每一個特性結構，請使用 fixedName。在下列範例中，類型 com.ibm.omnifind.types.Person 的每一個特性結構都會對映至索引中名爲「人員」的跨距。

  ```
  <indexBuildItem>
     <name>com.ibm.omnifind.types.Person</name>
     <indexRule>
  <style name="Annotation">
         <attribute name="fixedName" value="Person" />
       </style>
     </indexRule>
  </indexBuildItem>
  ```

  這樣會啓用「給我在人員名稱中包含上司的文件」之類的查詢。使用 XML 片段的查詢，表示式如下所示：@xmlf2::'<Person>Boss</Person>'
- 如果註解儲存了不同的實體，而您想要依據註解的特定特性值來利用不同的跨距存取這些實體，則請使用 nameFeature。在下列範例中，com.ibm.tt.EntityAnnotation 索引爲 person 或 organization 跨距，視名爲 type 的特性值而定。特性也可以是特性路徑。

```
<indexBuildItem>
<name>com.ibm.tt.EntityAnotation</name>
   <indexRule>
<style name="Annotation">
       <attribute name="nameFeature" value="type" />
     </style>
   </indexRule>
</indexBuildItem>
```

這樣會啓用「給我組織 WHO 的相關文件」(相對於英文的 "who" 一詞) 之類的查詢。使用有限的 XPath 語法，查詢表示式如下：

```
@xmlp::'/organization[ftcontains="WHO"]'
```

- 如果未使用上述任何屬性，則會使用 <indexBuildItem> 元素中註解類型的簡短名稱。這是預設值。例如：

```
<indexBuildItem>
<name>com.ibm.uima.tutorial.RoomNumber</name>
   <indexRule>
<style name="Annotation" />
<style name="Field" />
   </indexRule>
 </indexBuildItem>
```

此 <indexBuildItem> 元素會產生名爲 RoomNumber 的註解和欄位，並在其中輸入 com.ibm.uima.tutorial.RoomNumber 所涵蓋的文字。

<style name="Annotation" /> 元素

<style> 元素中的註解指定如何在企業搜尋中存取跨距資訊。除了容許使用 fixedName 及 nameFeature 屬性之外，此樣式還支援 <attributemappings> 元素。在這個元素中，可以將特性值對映至索引中結果跨距的屬性，以便後續在搜尋表示式中使用。

每一個對映都在個別的 <mapping> 元素中執行。<feature> 元素包含特性路徑，而 <indexName> 元素包含在索引中用來儲存 <feature> 值的屬性名稱。例如，

```
<mapping>
<feature>make/companyname</feature>
<indexName>company</indexName>
</mapping>
```

此 <mapping> 元素會將路徑 make/companyname 中的特性值直接儲存在索引屬性 company 中。

如果在文字分析期間使用的類型系統是複式的，其中包括許巢狀的特性結構，則將特性值對映至索引屬性特別有用。使用 <mapping> 元素時，會顯現相關屬性，可讓您在查詢中使用它們，而不必詳細地瞭解原始類型系統結構。

<style name="Field" /> 元素

<style> 元素中的欄位指定如何在企業搜尋中存取欄位資訊。除了 fixedName 及 nameFeature 屬性外，您還可以設定下列屬性。

parametric
: 如果設爲 true，則可以使用參數搜尋來搜尋欄位值，例如，#dosage:>100

fieldSearchable
: 如果設爲 true，則可以在搜尋中使用欄位值，例如 make:Bayer

returnable

如果設爲 true，則會在搜尋結果中傳回欄位及其值。

欄位資訊一律是可以搜尋的內容，亦即，可以在一般關鍵字搜尋中存取欄位資訊。

選用屬性 valueFeature 定義哪些特性值要當成欄位值。如果特性結構是註解，且未設定屬性，則會使用註解的涵蓋文字作爲欄位值。在範例中，

```
<indexBuildItem>
<name>com.ibm.omnifind.types.Date</name>
   <indexRule>
<style name="Field">
       <attribute name="fixedName" value="date"/>
       <attribute name="fieldSearchable"
value="true"/>
       <attribute name="returnable" value="true"/>
     </style>
<style name="Field">
       <attribute name="fixedName" value="hour"/>
       <attribute name="valueFeature"  value="hour"/>
       <attribute name="parametric" value="true"/>
     </style>
   </indexRule>
<filter syntax="FeatureValue">year="2005"</filter>
</indexBuildItem>
```

產生 com.ibm.omnifind.types.Date 的兩個欄位。一個欄位名稱爲 date 且含有涵蓋的文字，例如， 5:15pm。另一個欄位含有屬性 hour 的值。您可以在此使用 'hour::<17' 進行查詢。

<style name="Breaking" /> 元素

<style> 元素中的值 Breaking 不包括任何進一步元素。

建立 XML 檔之後，您必須將它上載至企業搜尋，然後使用企業搜尋管理主控台，來選取含有其他自訂分析選擇的共用分析結構到索引的對映檔。

相關概念

第 31 頁的『自訂分析結果的索引對映』
對文件集合執行自訂分析後， 您可以使用企業搜尋中的搜尋引擎，從儲存在共用分析結構 (以自訂分析演算法建立) 中的資訊建置索引。

第 27 頁的『特性路徑』
特性路徑可讓您存取共用分析結構中的特性值， 類似用來存取 XML 文件中 XML 元素的 XPath 陳述式。

相關參考

第 30 頁的『過濾器』
過濾器可用來限制共用分析結構到索引的對映檔，以及共用分析結構到資料庫的對映檔中的對映規則。只有在過濾條件爲眞時，分析結果才會加入索引或 JDBC 表格。

第 18 頁的『類型系統說明範例』
類型系統說明解釋在自訂分析中所使用的特性結構 (代表分析結果的基礎資料結構)。

所選分析結果的資料庫對映

在企業搜尋中分析文件之後，您可以將選取的文字分析結果儲存在具有 JDBC 功能的資料庫中。

此版本支援 DB2 Universal Database™ 8.2.2 版 (com.ibm.db2.jcc.DB2Driver 2.3 版) 或更高版本及 Oracle 10g (oracle.jdbc.driver.OracleDriver 1.0 版)。

對於 DB2 Universal Database 及 Oracle，您可以選擇將分析結果直接插入資料庫，或產生相等資料庫特定的載入檔及對應的 Script 以執行載入指令。

將分析結果對映至資料庫中的表格，可讓您在後續的商業情報處理程序步驟中使用此資訊，或直接存取文件中符合語意搜尋查詢的相關部分。

共用分析結構到資料庫的對映檔包含資料庫連線配置資訊，並說明要將哪些自訂分析結果要儲存在哪些表格及直欄中。對映檔中的表格及直欄名稱必須對應於在資料庫中建立的表格及直欄。

撰寫共用分析結構到資料庫的對映檔之後，您可以使用管理主控台將檔案上載至企業搜尋。

相關工作

第 39 頁的『建立共用分析結構到資料庫的對映檔』
若要將分析結果新增至資料庫，您必須建立共用分析結構到資料庫的對映檔，其中包含資料庫連線配置資訊，以及要將哪個自訂文字分析結果儲存在哪些資料庫表格及直欄的說明。

在資料庫中儲存分析結果

若要將選取的分析結果儲存在具有 JDBC 功能的資料庫中，您必須撰寫共用分析結構到資料庫的對映檔，其定義要將哪個分析結果儲存在資料庫中，且必要的 JDBC 驅動程式庫必須位於您在對映檔中所定義的路徑。

若要在具有 JDBC 功能的資料庫中儲存分析結果：

1. 決定要在資料庫中儲存哪些分析結果。建立資料庫，其中的表格含有適當資料類型的所有必要直欄。
2. 在 XML 編輯器中，使用資料庫配置資料撰寫共用分析結構到資料庫的對映檔，以及您要儲存的分析結果。爲了決定要將哪些分析結果併入對映檔，您必須知道處理文件時使用的基礎類型系統。
3. 將 JDBC 驅動程式庫放入索引節點上可以由企業搜尋系統存取的目錄中。
4. 使用企業搜尋管理主控台上載及選取對映檔。

使用載入檔案集

您可以直接將分析結果儲存在可處理 JDBC 的資料庫中，或者可以將處理作業配置爲使用載入檔集，並在稍後階段將資料載入資料庫。

使用載入檔案集具有下列優點：

- 總的來說，一組載入檔永遠不能大於作業系統支援的檔案大小上限
- 只要載入檔案集已滿，即可開始將資料載入資料庫中，而無需爲了避免檔案存取衝突而停止並重新啓動文件剖析器。

即使跨多個共用分析結構將文件分段，也會在文件層次上完成從一個載入檔案集切換至另一個載入檔案集。在已處理文件之後，如果現行載入檔案集中的某個載入檔超出

定義的限制，則會使用新的載入檔案集。這可保證載入檔案集的一致性。因爲主要表格中所有的項目都包含資料庫表格中的相符項目，所以在將一個載入檔案集的內容載入資料庫之後，資料模型會保持一致。

副檔名 .cur 會識別載入檔及 Script 檔。當已關閉載入檔集時，會重新命名檔案，使其具有副檔名 .dat。這表示該檔案可以在文件剖析器仍在執行時複製或移動到資料庫伺服器。

您可以指定載入檔的大小。當達到載入檔大小限制時，會啓動新的載入檔案集。在 <loadFile> XML 元素區段的共用分析結構到資料庫的對映檔中指定載入檔大小。將參數 loadFileSize 定義爲使用 <loadFileSize> 元素，並指定爲 10 MB <= loadFileSize <= 10240 (10MB <= loadFileSize <= 10GB)。<loadFileSize> 是選用元素。如果未設定任何值，則預設值是 1024MB (1GB)。

使用十位數的數字對檔案集中的單一載入檔進行編號，以識別哪個檔案屬於哪個載入檔案集。載入檔案集會在下列情況下關閉：

- 檔案集中的某個載入檔超出定義的大小限制
- 由於剖析器停止或發生錯誤而導致處理作業停止

如果重新啓動剖析器，則會使用新的載入檔案集，從處理作業停止的地方繼續執行它。

建立共用分析結構到資料庫的對映檔

若要將分析結果新增至資料庫，您必須建立共用分析結構到資料庫的對映檔，其中包含資料庫連線配置資訊，以及要將哪個自訂文字分析結果儲存在哪些資料庫表格及直欄的說明。

關於本作業

共用分析結構到資料庫的對映檔是 XML 格式。下列範例是以在治安報告實務範例中定義的類型系統爲基礎。

在範例中，只有治安報告和這些治安犯罪報告中出現的城市會新增至資料庫。此範例會顯示內建特性及 <constant> 元素對映的用法。

```
<?xml version="1.0" encoding="UTF-8"?>
<cas2JdbcConfiguration xmlns="http://www.ibm.com/uima/consumer/jdbc/100/xml">
   <databaseConnection>
<connectionUrl>db2://myMachine:myPort/myDatabase</connectionUrl>
<driver type="jdbc">com.ibm.db2.jcc.DB2Driver</driver>

     <driverLibraries>
<driverLibrary>C:\db2\db2jcc.jar</driverLibrary>
       <driverLibrary>C:\db2\db2jcc_license_cu.jar</driverLibrary>
<driverLibrary>C:\db2\db2jcc_license_cisuz.jar</driverLibrary>
     </driverLibraries>

     <authentication>
<username>myUser</username>
<password>myPassword</password>
     </authentication>

     <loadFile>
<loadFileDirectory>/home/cas2jdbc/load/</loadFileDirectory>
<loadFileSize>1048</loadFileSize>
<loadScript>/home/cas2jdbc/load/load.sh</loadScript>
```

```
     </loadFile>

   </databaseConnection>

   <cas2JdbcMappingSpec>
     <skipCondition>
<name>com.ibm.uima.tt.DocumentAnnotation</name>
<filter syntax="FeatureValue">toBeProcessed=0</filter>
     </skipCondition>

     <cas2JdbcMappings>
      <explicitMappings>
<explicitMappingRule applyToSubtypes="false">
<type>com.ibm.omnifind.types.PoliceReport</type>
<table>sample.policeReport</table>
        <featureMappings>
         <featureMapping>
<feature>uniqueId()</feature>
<column>policeReportId</column>
         </featureMapping>
         <featureMapping>
<feature>location/uniqueId()</feature>
<column>crimeLocationId</column>
         </featureMapping>
        </featureMappings>
<filter syntax="FeatureValue">location/coveredText()="Los Angeles"</filter>
       </explicitMappingRule>
      </explicitMappings>

      <implicitMappings>
<implicitMappingRule applyToSubtypes="false">
<type>com.ibm.omnifind.types.City</type>
<table>sample.City</table>
        <featureMappings>
         <featureMapping>
<feature>uniqueId()</feature>
<column>crimeLocationId</column>
         </featureMapping>
         <featureMapping>
<feature>coveredText()</feature>
<column>cityName</column>
<length>150</length>
         </featureMapping>
         <featureMapping>
<constant>USA</constant>
<column>country</column>
         </featureMapping>
        </featureMappings>
       </implicitMappingRule>
      </implicitMappings>

     </cas2JdbcMappings>
   </cas2JdbcMappingSpec>
</cas2JdbcConfiguration>
```

程序

若要建立共用分析結構到資料庫的對映檔，請：

1. 建立 XML 檔。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。對映檔的 XSD 綱目稱爲 CasToJDBCMapping.xsd，且內含於企業搜尋安裝的 *ES_INSTALL_ROOT*/packages/uima/configuration_xsd/ 中。

2. 在 <cas2JdbcConfiguration xmlns="http://www.ibm.com/uima/consumer/jdbc/100/xml"> 元素中併入對映。名稱空間 (在 xmlns 屬性中指定) 必須如所示範例一模一樣。
3. 新增包含所有資料庫連線配置資訊的 <databaseConnection> 元素，以及說明儲存在資料庫或載入檔的分析結果對映規則的 <cas2JdbcMappingSpec> 元素。
4. 將下列元件元素新增至 <databaseConnection> 元素：
   - 必備元素：<connectionUrl> 元素。此元素含有資料庫連線 URL。您可以視 JDBC 驅動程式實作而定，以本端或遠端存取資料庫。
   - 必備元素：<driver> 元素。此元素含有 JDBC 驅動程式類別的名稱，例如，com.ibm.db2.jcc.DB2Driver (適用於 DB2®) 或 oracle.jdbc.driver.OracleDriver (適用於 Oracle)。
   - 必備元素：<driverLibraries> 元素。此元素列出驅動程式庫。每一個程式庫都列示在 <driverLibrary> 元素中。程式庫是在 DB2 或 Oracle 安裝目錄中。若爲 DB2，程式庫是 c:\your_db2_dir\db2jcc.jar、c:\your_db2_dir\db2jcc_license_cu.jar 及 c:\your_db2_dir\db2jcc_license_cisuz.jar。若爲 Oracle，包含的程式庫是 c:\your_oracle_dir\classes12.zip。

     確定驅動程式庫始終與 DB2 Applet 伺服器位於相同的維護層次。
   - 必備元素：<authentication> 元素。此元素含有資料庫的使用者名稱及密碼。
   - 選用元素：<loadFile> 元素。此元素包含下列元件元素：
     - <loadFileDirectory> 元素中的載入檔目錄。
     - 選用元素：<loadFileSize> 元素中的載入檔大小。載入檔大小限制是 10 <= loadFileSize <= 10240 (10MB <= loadFileSize <= 10GB)。如果未定義值，則預設值是 1024 MB (1GB)。
     - <loadScript> 元素中的載入 Script 名稱。

     如果您沒有指定 <loadFile> 元素，則可利用 JDBC 將所有資料直接儲存在資料庫中。

     當您使用資料庫特有的載入檔及 Script 時，您還必須新增所有資料庫配置參數。
5. 將下列元件元素新增至 <jdbcMappingSpec> 元素：
   - 選用元素：<skipCondition> 元素。如果沒有定義略過條件，則會處理所有文件。

     ```
     <skipCondition>
       <name>com.ibm.uima.tt.DocumentAnnotation</name>
       <filter syntax="FeatureValue">toBeProcessed=0</filter>
     </skipCondition>
     ```

     在範例中，將不考慮註解類型爲 com.ibm.uima.tt.DocumentAnnotation 且特性 toBeProcessed 設爲 0 的文件。
   - <cas2JdbcMappings> 元素，顯示要將哪些類型及特性對映至哪些資料庫表格及直欄。元素含有明確及隱含的對映區段。
6. 新增 <explicitMappings> 元素。這是必備元素。它必須具有一或多個定義明確對映的 <explicitMappingRule> 元素，且只能針對註解類型及其次類型定義。 如果對映定義於明確對映區段，則符合對映定義的所有註解都會儲存在資料庫中。

7. 選用元素：新增 <implicitMappings> 元素。此元素支援所有特性結構類型。如果此元素存在，則必須至少包含一個 <implicitMappingRule> 元素。 只有在另一個符合明確或隱含對映規則的註解參照相符註解類型時，才能將定義於隱含對映區段的對映加入資料庫。

   隱含對映的目的是要讓您只儲存出現在特定環境定義的分析結果。比方說，如果註解類型 com.ibm.omnifind.types.City 的對映是隱含的，則只有明確對映區段中 com.ibm.omnifind.types.PoliceReport 對映定義所參照的城市會儲存在資料庫中。這表示只有治安報告所提到的城市會新增至資料庫。

   如果有適用於 City 註解的明確對映規則，則所有城市都會新增至資料庫。在這兩種情況下，如果某一城市有多個治安報告參照，也只會加入資料庫一次。

8. <explicitMappingRule> 及 <implicitMappingRule> 元素必須包含屬性 applyToSubtypes，如果此屬性設爲 true，則不僅會儲存列示在 <type> 元素中的特性結構，還會儲存從該結構所衍生出來的所有特性結構。將下列元件元素新增至 <explicitMappingRule> 及 <implicitMappingRule> 元素：
   - <type> 元素，包含特性結構類型。
   - <table> 元素，包含資料庫綱目及表格名稱。語法遵循規則 schema.table_name，或如果未定義綱目，則只有 table_name。
   - <featureMappings> 元素，包含一或多個 <featureMapping> 元素或者一個 <containerMapping> 元素。
   - 選用元素：<filter> 元素，包含每次對映規則相符時評估的條件。如果條件評估爲 true，則特性結構的註解會儲存在資料庫中。在範例中，只有處理洛杉磯犯罪的治安報告會儲存在資料庫中。
9. <featureMapping> 元素元件結構會隨著您是否對映特性或常數而改變。

   如果您要對映特性或特性路徑，則元件元素包括：
   - <feature> 元素，含有特性名稱。必須定義類型元素中特性結構的特性。您也可以使用特性路徑建構或任何系統定義的內建特性。
   - 選用元素：<length> 元素，其字串長度定義於指定的資料庫直欄。較長的字串會被截斷。
   - <column> 元素，含有要儲存特性值的直欄名稱。未在任何特性對映中使用的資料庫直欄會使用資料庫配置的預設值 (通常是空值)。

     請確定特性元素的值儲存在適當類型的直欄中。下表顯示哪些 UIMA 類型符合哪些資料庫類型。

表 3. UIMA 類型及相對應的資料庫類型之間的對映

UIMA 類型或內建特性	建議的 DB2 資料類型	建議的 Oracle 資料類型
Float	REAL	FLOAT
String	VARCHAR	VARCHAR2
Integer	INTEGER	INTEGER
uniqueId(), uniqueParentId()	CHAR(27)	CHAR(27)
objectId(), parentId()	CHAR(16)	CHAR(16)
docTimestamp()	BIGINT	LONG
fsId()	INTEGER	INTEGER

若爲常數，元件特性對映元素如下：

- <constant> 元素，包含常數值。
- <column> 元素，含有要新增常數值的直欄名稱。

10. <containerMapping> 元素，包含儲存區類型特性 (陣列或清單) 的對映。此元素只適用於儲存區類型。有下列元件元素：
    - <feature> 元素，含有特性名稱。您也可以使用特性路徑建構或任何系統定義的內建特性。
    - <table> 元素，包含資料庫綱目及表格名稱。語法遵循規則 schema.table_name，或如果未定義綱目，則只有 table_name。
    - 一或多個 <featureMapping> 元素，其中包含特性結構名稱及要新增功能的直欄名稱。
11. 使用提供的綱目來儲存及驗證 XML 檔。

建立 XML 檔之後，您必須將它上載至企業搜尋，然後使用企業搜尋管理主控台，來選取具有其他自訂分析選擇的共用分析結構到資料庫的對映檔。

相關概念

第 37 頁的『所選分析結果的資料庫對映』
在企業搜尋中分析文件之後，您可以將選取的文字分析結果儲存在具有 JDBC 功能的資料庫中。

第 27 頁的『特性路徑』
特性路徑可讓您存取共用分析結構中的特性值， 類似用來存取 XML 文件中 XML 元素的 XPath 陳述式。

相關參考

第 30 頁的『過濾器』
過濾器可用來限制共用分析結構到索引的對映檔，以及共用分析結構到資料庫的對映檔中的對映規則。只有在過濾條件爲眞時，分析結果才會加入索引或 JDBC 表格。

第 28 頁的『內建特性』
內建特性是預先定義的特性名稱，這些名稱具有特殊的語意。 它們可以用來存取特性結構本身沒有的資訊，例如，特性結構的類型或註解的涵蓋文字。 它們可以當成特性路徑中的最後一個或唯一的元素使用。

第 18 頁的『類型系統說明範例』
類型系統說明解釋在自訂分析中所使用的特性結構 (代表分析結果的基礎資料結構)。

儲存區類型對映

儲存區類型是共用分析結構中的其中一個內建陣列或清單類型。儲存區類型對映是將陣列或清單值對映至關聯式資料庫的一種方式。

有兩種方法可以處理共用分析結構到資料庫對映檔中的儲存器類型。方法之一是使用已定義的內建特性建構及一般鏈結表格，其中所含的陣列或清單是特性對映規則的值。因爲不同的陣列或清單都儲存在相同的鏈結表格中，所以表格無法指出儲存資訊的關係。

在第二種方法中，使用 <containerMapping> 元素定義的鏈結表格定義明確地表示您要擁有的指定資訊之間的關係。

下面範例顯示一般鏈結表格對映可能的樣子。在治安報告和嫌犯之間有 n:m 關係，表示某一嫌犯會在一或多個治安報告中提及，且一份治安報告可以提及多個嫌犯。

範例中的一般 sample.fsarray 表格是治安報告和嫌犯之間的鏈結表格。如果除了特性類型 com.ibm.omnifind.types.FSArray 的 com.ibm.omnifind.types.PoliceReport 外，還有其他對映類型，也會對映至這個表格。您還是可以查詢表格以瞭解治安報告和嫌犯之間的正確關係，但是，您不能只是查看表格就斷定其中含有治安報告和可能嫌犯之間的關係或鏈結。

```
<cas2JdbcMappings>
  <explicitMappings>
<explicitMappingRule applyToSubtypes="false">
<type>com.ibm.omnifind.types.PoliceReport</type>
<table>sample.policeReport</table>
      <featureMappings>
        <featureMapping>
<feature>uniqueId()</feature>
<column>policeReportId</column>
        </featureMapping>
        <featureMapping>
<feature>knownSuspects/uniqueId()</feature>
<column>suspectArrayId</column>
        </featureMapping>
        <featureMapping>
<feature>location/cityName</feature>
<column>city</column>
        </featureMapping>
      </featureMappings>
    </explicitMappingRule>
  </explicitMappings>

  <implicitMappings>
<implicitMappingRule applyToSubtypes="false">
<type>com.ibm.omnifind.types.Suspect</type>
<table>sample.suspect</table>
      <featureMappings>
        <featureMapping>
<feature>uniqueId()</feature>
<column>suspectID</column>
        </featureMapping>
        <featureMapping>
<feature>surName</feature>
<column>lastName</column>
        </featureMapping>
        <featureMapping>
<feature>description</feature>
<column>description</column>
        </featureMapping>
      </featureMappings>
    </implicitMappingRule>

<implicitMappingRule applyToSubtypes="false">
<type>uima.cas.FSArray</type>
<table>sample.fsarray</table>
      <featureMappings>
        <featureMapping>
<feature>uniqueId()</feature>
<column>arrayId</column>
        </featureMapping>
        <featureMapping>
<feature>[:index]</feature>
<column>arrayIndex</column>
        </featureMapping>
        <featureMapping>
<feature>[]/uniqueId()</feature>
```

```
<column>suspectId</column>
        </featureMapping>
      </featureMappings>
    </implicitMappingRule>
  </implicitMappings>

</cas2JdbcMappings>
```

下面顯示以上述一般對映規則爲基礎的資料庫表格。

表 4. sample.policeReport 表格

policeReportId	suspectArrayId	city
aaa...1	**bbb...1**	Springfield
aaa...2	**bbb...2**	Ladysmith

表 5. sample.fsarray 表格

arrayId	arrayIndex	suspectId
bbb...1	1	*ccc...1*
bbb...1	2	*ccc...2*
bbb...2	1	*ccc...3*

表 6. sample.suspect 表格

suspectID	lastname	說明
ccc...1	Brown	Dark complexion
ccc...2	Smith	Wears glasses
...	...	...

範例顯示特性結構陣列的對映。您也可以將此類型的對映套用於 StringArray、IntegerArray 及 FloatArray。如果併入這些簡式值陣列的對映規則，請以 `[]` 取代 `[]/uniqueId()`。

相同的一般表格方法可以用於特性結構清單，以及簡式類型清單 (StringList、IntegerList 及 FloatList)。

更簡單的處理關係方法是使用明確的儲存區對映元素，它定義陣列或清單所含元素的疊代。

下面範例中的對映指出明確的鏈結表格。而在治安報告及嫌犯之間，再次存在 n:m 關係。但是，這次 `sample.reports_suspects` 表格是治安報告和嫌犯之間的鏈結表格。

在這種方式中，您不必考慮處理陣列 ID，或清單類型的頭尾項目對映。鏈結表格含有一個明確的關係。

```
<cas2JdbcMappings>
  <explicitMappings>
<explicitMappingRule applyToSubtypes="false">
<type>com.ibm.omnifind.types.PoliceReport</type>
<table>sample.policeReport</table>
      <featureMappings>
        <featureMapping>
<feature>uniqueId()</feature>
<column>policeReportID</column>
        </featureMapping>
```

```
        <featureMapping>
<feature>location/cityName</feature>
<column>city</column>
        </featureMapping>
        <featureMapping>
<feature>knownSuspects</feature>
          <containerMapping>
<table>sample.reports_suspects</table>
            <featureMapping>
              <feature>com.ibm.omnifind.types.PoliceReport
/objectId()</feature>
<column>policeReportId</column>
            </featureMapping>
            <featureMapping>
<feature>knownSuspects/[]/objectId()</feature>
<column>suspectId</column>
            </featureMapping>
          </containerMapping>
        </featureMapping>
      </featureMappings>
    </explicitMappingRule>
  </explicitMappings>

  <implicitMappings>
<implicitMappingRule applyToSubtypes="false">
<type>com.ibm.omnifind.types.Suspect</type>
<table>sample.suspect</table>
      <featureMappings>
        <featureMapping>
<feature>objectId()</feature>
<column>suspectID</column>
        </featureMapping>
        <featureMapping>
<feature>surName</feature>
<column>lastName</column>
        </featureMapping>
        <featureMapping>
<feature>description</feature>
<column>description</column>
        </featureMapping>
      </featureMappings>
    </implicitMappingRule>
  </implicitMappings>

</cas2JdbcMappings>
```

`<containerMapping>` 元素可用來定義陣列所含元素的疊代。在範例中，`sample.reports_suspects` 鏈結表格含有 `policeReportId` 及 `suspectId` 直欄的鏈結。請勿將 `<containerMapping>` 元素巢狀化。

下面顯示以明確鏈結表格對映規則爲基礎的資料庫表格。

表 7. sample.policeReport 表格

policeReportId	**city**
aaa...1	Springfield
aaa...2	Ladysmith

表 8. sample.reports_suspect 表格

policeReportId	**suspectId**
bbb...1	*ccc...1*

表 8. *sample.reports_suspect 表格 (繼續)*

policeReportId	suspectId
bbb...2	*ccc...2*
...	...

表 9. *sample.suspect 表格*

suspectID	lastname	說明
ccc...1	Brown	Dark complexion
ccc...2	Smith	Wears glasses
...	...	...

相關參考

第 28 頁的『內建特性』
內建特性是預先定義的特性名稱，這些名稱具有特殊的語意。 它們可以用來存取特性結構本身沒有的資訊，例如，特性結構的類型或註解的涵蓋文字。 它們可以當成特性路徑中的最後一個或唯一的元素使用。

擷取文件中符合語意搜尋查詢的部分

藉由將相關的特性結構同時對映至索引與資料庫，並在語意搜尋查詢中指定跨距，您可以只擷取文件中完全符合查詢的部分。

若要存取搜尋結果中特定註解類型的所有實例 (例如，若要取得所有人員)，請併入註解類型的欄位樣式對映，並在共用分析結構到索引的對映檔中將它標示爲可傳回。例如：

```
<indexBuildItem>
 <name>com.ibm.omnifind.types.Person</name>
 <indexRule>
<style name="Annotation"/>
<style name="Field">
<attribute name="returnable" value="true"/>
  </style>
 </indexRule>
</indexBuildItem>
```

在此範例中，類型 `com.ibm.omnifind.types.Person` 的註解會對映要企業搜尋索引中名稱爲「人員」的跨距，在語意搜尋時可以在其中存取它們。此外，註解的涵蓋文字 (例如，完整的人員名稱) 會儲存爲可傳回的欄位。若要擷取這些註解值，請對搜尋查詢 (關鍵字或語意) 傳回的每一個結果物件呼叫 getFields(″Person″)。此方法會傳回含有註解值的「字串」陣列，在此案例中，會傳回人員名稱。

然而，這種方式會傳回指定註解類型的所有實例，且如果您要將結果處理程序限制於完全符合查詢的文件時，則不適用。例如，文件可能會提到五個人。然而，在語意搜尋查詢 `'<sentence><person/>IBM</sentence>` 中，使用者只關注與 IBM 一詞出現在同一個句子中的人員。使用者並不想知道其他人。

若要存取並處理完全符合查詢的特性結構：

1. 使用註解對映樣式，將相關的特性結構類型對映至企業搜尋索引。 例如：

```
<indexBuildItem>
 <name>com.ibm.omnifind.types.Person</name>
 <indexRule>
<style name="Annotation"/>
 </indexRule>
</indexBuildItem>
```

2. 將相關的特性結構類型對映至 JDBC 表格。在對映的程序中，您必須併入文件 URI 和特性結構 ID 的兩個直欄。雖然可以將所有特性結構類型對映至相同的資料庫表格，但應將每一個類型對映至不同的表格。 例如：

```
<explicitMappingRule applyToSubtypes="false">
<type>com.ibm.omnifind.types.Person</type>
<table>sample.person</table>
        <featureMappings>
         <featureMapping>
<feature>objectId()</feature>
<column>primaryId</column>
         </featureMapping>
<!-- Contains the covered text of the annotation-->
         <featureMapping>
<feature>coveredText()</feature>
<column>personName</column>
         </featureMapping>
<!-- Other mapping go in here-->
         <!-- To access the relevant person annotations in the query result-->
         <featureMapping>
<feature>docUri()</feature>
<column>docUri</column>
         </featureMapping>
         <featureMapping>
<feature>fsId()</feature>
<column>annotationId</column>
         </featureMapping>
        </featureMappings>
       </explicitMappingRule>
```

3. 搜索、剖析及索引文件。
4. 擷取符合查詢的實例 ID。在搜尋及索引 API (SIAPI) 中，這些實例稱爲目標元素。目標元素指定要傳回的輸入跨距。它的定義如下：
   - 在 XML 片段中，利用附加在前面的 # 記號來識別目標元素。在 XML 片段查詢中只容許使用一個 # 記號，該記號可以出現在任何位置。例如：`$xmlf2::'<sentence><#person/>IBM</sentence>'`
   - 預設在 XPath 中，目標元素是 XPath 表示式中的最後一個欄位。
   - 利用方法 Result.getProperty(″TargetElement″) 來存取這些實例。傳回的內容是字串連接，其中含有所有出現 ID 並以空格區隔。內容中的每一個出現項目都可以轉換成整數值。
5. SIAPI 不會自行傳回特性結構，只會傳回其出現 ID。這些 ID 對應於儲存在資料庫表格的 fsId() 值。若要擷取這些實例及相關資訊，您的應用程式必須：
   a. 依據目標元素的跨距名稱，選取正確的資料庫表格。在範例中，應用程式含有從人員到 sample.Person 表格的對映。此資訊是從共用分析結構到索引的對映檔 (產生跨距名稱)，以及共用分析結構到資料庫的對映檔 (產生表格名稱) 推斷出來的。
   b. 針對搜尋結果中的每一個結果物件：
      1) 剖析 Result.getProperty (″TargetElement″) 傳回的字串，以尋找出現 ID。

2) 使用結果 URI (可以使用 Result.getDocumentId() 來存取) 作爲 docUri 直欄中的值，以及出現 ID 作爲 annotationId 直欄中的值，以發出表格的 SELECT 陳述式。直欄名稱需視對映的檔案而定。直欄名稱取自前一個範例。

傳回的列包含特性結構的儲存資訊 (例如，涵蓋的文字)，或特性結構的特定屬性 (如「姓氏」或「出生地」)。

請確定資料庫的更新與企業搜尋中的索引更新同步。如果資料庫含有過期的資訊 (例如，因爲您使用資料庫載入檔，且沒有更新資料庫，但已重新整理或重組了索引)，則部分出現 ID 可能不會存在於資料庫中。企業搜尋只會在索引中保留最新文件版本的記錄。因此，出現 ID 只適用於最新的文件。

如果在相同的資料庫表格中儲存了同一份文件的多個版本，則可能會有多列符合相同的出現 ID，每一列代表不同的文件版本。在此情況下，您必須定義文件版本直欄，並使用應用程式邏輯或內建特性 (如 docTimestamp()) 來輸入資料。這樣，您就可以過濾結果，只取得最新的文件版本。

相關概念

第 50 頁的『語意搜尋查詢字詞』
語意搜尋查詢字詞是以不透明字詞來傳達。

相關工作

第 32 頁的『建立共用分析結構到索引的對映檔』
使用共用分析結構到索引的對映檔，您可以決定要索引共用分析結構中的哪些分析結果以啓用搜尋。

第 39 頁的『建立共用分析結構到資料庫的對映檔』
若要將分析結果新增至資料庫，您必須建立共用分析結構到資料庫的對映檔，其中包含資料庫連線配置資訊，以及要將哪個自訂文字分析結果儲存在哪些資料庫表格及直欄的說明。

語意搜尋應用程式

四種類型的文件資訊儲存在企業搜尋索引中，您可以使用搜尋及索引 API (SIAPI) 介面，在搜尋應用程式中查詢此資訊。

四種不同類型的資訊包括：

- 在文件中找到的文字字組，例如，*電腦軟體*之類的詞組。
- 跨距名稱，例如，包括 <author>James</author> 的 XML 文件，會產生跨距 <author>。
- 屬性名稱，例如，包括 <author countryOfBirth=USA>James</author> 的 XML 文件，會產生屬性 ″countryOfBirth″。
- 屬性值，例如，USA 是屬性 ″countryOfBirth.″ 的值。

SIAPI 查詢語言包含語意搜尋查詢字詞。該字詞指定細枝型樣。細枝是有葉子的小樹。每一片葉子都代表四種類型的資訊 (文字字組、跨距名稱等等)。樹的內部節點指定它們在文件中的出現如何彼此關聯。指定關係的內部節點有五種類型：

- and
- or
- not
- in_the_span_of

- attribute_in_the_span_of

如果文件中有出現葉子，則可用來滿足指定的語意搜尋字詞，並遵守內部節點指定的限制 (已定義的關係)。

語意搜尋查詢字詞有助於擷取更好品質的文件。您現在不只能利用單字和註解的 Boolean 組合來搜尋。還可以擷取 (舉例來說) *James* 出現在跨距具名作者，或 *ibm* 及*搜尋* 詞彙出現在相同句子的文件。

語意搜尋查詢字詞

語意搜尋查詢字詞是以不透明字詞來傳達。

有兩種語法形式可用來在搜尋及索引 API (SIAPI) 中表示不透明字詞：

- XML 片段
- 限制的 XPath

XML 片段查詢字詞類似 XML 文件中平衡良好的片段。 XML 片段查詢字詞的字首是不透明字詞符號 `@xmlf2::`，後面接著以單引號括住的 XML 片段表示式 ('...').

然而，限制的 XPath 查詢字詞字首是 `@xmlxp::`，後面接著以單引號括住的 XPath 查詢 ('...').

在搜尋及索引 API (SIAPI) 介面中使用一般查詢字詞時，每一個字詞都會有外觀修飾元：

加號 (+)
字詞必須出現。

字首 =
字詞必須完全相符。

字首 ~ 字元
考慮查詢字詞的同義字。

字尾 ~ 字元
考慮詞形與查詢字詞相同的單字。

記號 (#)
強調顯示字詞。

下列範例顯示 XML 片段查詢。

@xmlf2::'<City>Springfield</City>'
尋找跨距 (註解)「城市」含有字串 Springfield 的文件。

@xmlf2::'<Person gender="female"/>'
尋找其中註解女性人員的文件。

@xmlf2::'<Person><.or><@gender>female</@gender>
<@title>Mrs</@title><@title>Ms</@title></.or></Person>'
尋找依據性別或職稱指定人員爲女性的文件。

@xmlf2::'<Person gender="male" role="suspect"/>
<PoliceReport><@crimeDescription><.or>robbery theft</.or>-accident
</@crimeDescription></PoliceReport> <City>Springfield<.or>

<@district>Brynston</@district><@district>Brooklyn</@district></.or></City>'

尋找指定男性爲嫌犯的文件，以及使用屬性 crimeDescription 中的字串 *robbery* 或 *theft* (而不是字串 *accident*) 賦予屬性的 PoliceReport 註解。文件也必須包含覆蓋文字單字 *Springfield* 的城市註解，它是使用 *Brynston* 或 *Brooklyn* 特區賦予屬性的註解。

相對應的 XPath 查詢含有下列結構：

@xmlxp::'//City ftcontains ("Springfield")'

尋找跨距 (註解)「城市」含有字串 *Springfield* 的文件。

@xmlxp::'//PoliceReport[City ftcontains("Springfield")]'

尋找在跨距 PoliceReport 中包括含有字串 *Springfield* 之跨距 (註解)「城市」的文件。

@xmlxp::'//Person[@gender="female" or @title ftcontains("Ms") or @title ftcontains("Mrs")]'

尋找其中註解女性人員的文件。在性別屬性中，值必須完全相符，但是對於標題屬性，*Ms* 及 *Mrs* 不需要完全符合屬性值。

搜尋應用程式中的同義字支援

您可以藉由搜尋包含查詢字詞同義字的文件，來擴充搜尋結果。

同義字通常包含多字詞彙，如 *WebSphere Information Integrator OmniFind* 之類的產品名稱。同義字定義檔所含的多字詞彙可在使用者查詢中正確地識別，而不需要以引號括住。

企業搜尋的「搜尋及索引 API (SIAPI)」支援多種方法，以便使用者搜尋查詢字詞的同義字：

- SIAPI 查詢語法支援以波浪符號 (~) 運算子，來延伸同義字。如果使用者在查詢詞彙開頭加上此運算子，則會延伸到該字詞的同義字。例如，查詢 `~WAS` 會傳回討論 WebSphere Application Server，以及此縮寫的其他任何同義字的文件。
- 搜尋應用程式內可以使用 SIAPI 同義字延伸介面，來啓用同義字延伸。查詢詞彙可以自動擴充爲納入同義字，或者，搜尋應用程式可能包含選項，讓使用者指定搜尋結果是否傳回查詢詞彙的同義字。

  在自動擴充同義字期間，會針對所有查詢字組執行同義字查閱。搜尋結果會傳回含有查詢詞彙或其同義字的文件。SIAP 還會支援產生送出查詢的同義字擴充清單。

請勿針對利用 n-gram 斷詞法處理的文字使用同義字支援。

建立同義字的 XML 檔

若要將企業搜尋中的查詢擴增爲包括查詢字詞的同義字，必須在 XML 檔中，指定可以彼此做爲同義字的字組。此 XML 檔可用來建置二進位定義檔，讓您上載至企業搜尋並指派給適當的集合。

關於本作業

列出同義字的 XML 檔必須符合特定的綱目。以下是同義字的範例 XML 檔：

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups xmlns="http://www.ibm.com/of/822/synonym/xml">
  <synonymgroup>
    <synonym>Think Pad</synonym>
    <synonym>Notebook</synonym>
    <synonym>Notebooks</synonym>
  </synonymgroup>
  <synonymgroup>
    <synonym>WebSphere Application Server</synonym>
    <synonym>WAS</synonym>
  </synonymgroup>
</synonymgroups>
```

限制

您必須將彼此是同義字的字組 (`<synonym>` 元素)，組織在 `<synonymgroup>` 元素中。同義字可以包括空格字元，但不可以包括標點字元，例如逗點 (,) 或垂直線 (|)，因爲這些字元可能會影響企業搜尋查詢語法。

您必須列舉所有新增爲同義字的可能詞彙變化，例如字組的單數和複數形式。您不需要列舉字詞的正規化，例如移除重音或曲音 (企業搜尋會自動處理正規化)，也不需要併入字詞的大寫及小寫變體。比方說，如果要將詞彙 météo 加入爲同義字，並不需要將詞彙 METEO 也加入。

程序

若要建立同義字清單，供進行企業搜尋：

1. 建立 XML 檔。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。XML 檔的 XSD 綱目稱爲 synonyms.xsd，且內含於企業搜尋安裝的 *ES_INSTALL_ROOT*/packages/uima/configuration_xsd/ 中。
2. 新增 `<synonymgroup>` 元素，然後針對每一個被視爲同義字群組中其他字組之同義字的單字插入 `<synonym>` 元素。

   請確定在 `<synonymgroups xmlns="http://www.ibm.com/of/822/synonym/xml">` 元素中包括對映。名稱空間 (在 xmlns 屬性中指定) 需要如所示範例一模一樣。
3. 重複之前的步驟，直到指定好要用來在企業搜尋集合中搜尋文件的所有同義字。
4. 儲存並結束 XML 檔。

建立好 XML 檔之後，必須將該檔案轉換爲同義字定義檔，才能新增到企業搜尋系統。

建立同義字定義檔

在 XML 檔中建立或更新同義字清單之後，您必須將 XML 檔轉換爲二進位同義字定義檔。

關於本作業

若要建立同義字定義檔，請使用 WebSphere II OmniFind Edition 隨附的指令行工具，名爲 essyndictbuilder。這個工具是在 *ES_INSTALL_ROOT*/bin 目錄中。

工具的輸入是列出同義字的 XML 檔，工具的輸出就是同義字定義檔。定義檔必須具有字尾 `.dic`。例如，`c:\mydictionaries\products.dic`。

這兩個檔案的預設位置，就是呼叫 Script 的目錄。如果已存在相同名稱的定義檔，Script 會產生錯誤。

`.dic` 在企業搜尋中的大小上限是 8 MB。

程序

若要建立企業搜尋的同義字定義檔：

1. 以企業搜尋管理者的身分登入索引伺服器。 此使用者 ID 是在安裝 WebSphere II OmniFind Edition 時指定。
2. 輸入下列命令，其中的 *`XML_file`* 是含有同義字清單之 XML 檔案的完整路徑，*`DIC_file`* 是同義字定義檔的完整路徑。

   AIX®、Linux® 或 Solaris：`essyndictbuilder.sh` *`XML_file DIC_file`*
   Windows：`essyndictbuilder.bat` *`XML_file DIC_file`*

建立好同義字定義檔後，使用企業搜尋管理主控台，將定義檔新增到企業搜尋系統，並將它與一或多個集合連結。

只有產生的 .dic 檔案，才會上載到企業搜尋系統。請確定將原始的 XML 檔放在實施存取控制的環境中，並且確定您正常地備份檔案。您以後要更新同義字定義檔時，將會需要這個 XML 檔。

自訂停用字定義檔

您可以定義要從查詢中移除的企業特有詞彙，以增進搜尋關聯。

企業搜尋提供兩種停用字支援：

- 語言特有的停用字識別，可以從多字查詢中移除所有常用的通用字，如 *a* 及 *the*。使用者無法修改每一個語言的停用字定義檔。這個停用字識別會自動在所有查詢中執行，以增進搜尋關聯性。
- 使用者定義或自訂停用字識別，可以從查詢中移除企業特有詞彙。這個停用字定義檔是由管理者定義，其中只含特殊詞彙。使用者定義的停用字定義檔並不會取代含有通用字的企業搜尋語言特有停用字定義檔。使用者定義的停用字定義檔與語言無關。

使用者定義的停用字通常含有多字詞彙，如 *WebSphere Information Integrator OmniFind* 之類的產品名稱。停用字定義檔所含的多字詞彙可在使用者查詢中正確地識別，而不需要以引號括住。

德文中的複合字詞可以在查詢中正確識別。複合字詞是當作單一字使用之兩個或多個字的組合。詞彙化複合，如 *Reisebüro* (旅行社) 不會被視爲複合。

會分解查詢中的複合字詞爲組成複合的個別字詞。如果構成複合字詞的任何個別詞彙出現在停用字定義檔中，則不會從查詢中移除該複合字詞。

例如，查詢字詞 `Versicherungspolice` (保險原則) 會傳回含有複合字詞 *Lebensversicherungspolice* (人壽保險原則) 及 *Haftpflichtversicherungspolice* (第三方保險原則) 的文件。即使單字 *Police* 列示在停用字定義檔中，也不會從查詢中移除複合查詢字詞 `Versicherungspolice`。

您必須在 XML 檔中列出企業特有詞彙，然後將該檔案轉換成停用字定義檔，才能將它新增至企業搜尋系統。

您可以在企業搜尋管理主控台上選取要使用哪一個停用字定義檔。您可以爲每一個集合選取一個停用字定義檔。而一個停用字定義檔可由數個集合共用。

建立停用字的 XML 檔

若要從查詢中移除企業特有的詞彙，您必須指定哪些單字在 XML 檔中定義爲停用字。

關於本作業

列出停用字的 XML 檔必須符合 XML 文件中指定的特定綱目。以下是停用字的 XML 檔範例：

```
<?xml version="1.0" encoding="UTF-8"?>
<stopWords xmlns="http://www.ibm.com/of/83/stopwordbuilder/xml">
   <stopWord>OmniFind Edition</stopWord>
   <stopWord>WAS</stopWord>
   <stopWord>...</stopWord>
</stopWords>
```

限制

停用字可以包含空格字元，但不能含有標點字元，如逗點 (,) 或垂直線 (|)，因爲這些字元可能會影響企業搜尋查詢語法。

您不需要列舉詞彙的正規化，例如移除重音或曲音 (企業搜尋會自動處理正規化)。比方說，如果您要將詞彙 météo 併入爲停用字，則不需要同時加入詞彙 METEO。

程序

若要建立企業搜尋的停用字清單：

1. 建立 XML 檔。 若要避免 XML 語法錯誤，請使用可以驗證 XML 的 XML 編輯器或 XML 編寫工具。 XML 檔的 XSD 綱目稱爲 stopWords.xsd，且內含於企業搜尋安裝的 *ES_INSTALL_ROOT*/packages/uima/configuration_xsd/ 中。
2. 針對每一個被視爲停用字的單字，新增 `<stopWord>` 元素。

   請務必在 `<stopWords xmlns="http://www.ibm.com/of/83/stopwordbuilder/xml">` 元素中併入對映。名稱空間 (在 xmlns 屬性中指定) 需要如所示範例一模一樣。
3. 重複之前的步驟，直到已指定使用者搜尋企業搜尋集合時要從查詢中移除的所有停用字爲止。
4. 儲存並結束 XML 檔。

建立 XML 檔後，您必須將它轉換成停用字定義檔，然後才能新增至企業搜尋系統。

建立停用字定義檔

在 XML 檔中建立或更新使用者定義的停用字清單後，您必須將 XML 檔轉換成停用字定義檔。

關於本作業

若要建立停用字定義檔，請使用 WebSphere II OmniFind Edition 隨附的指令行工具，名爲 esstopworddictbuilder。這個工具是在 *ES_INSTALL_ROOT*/bin 目錄中。

工具的輸入是列出停用字的 XML 檔，而工具的輸出就是停用字定義檔。定義檔必須具有字尾 `.dic`。例如，`c:\mydictionaries\productstopwords.dic`。

這兩個檔案的預設位置，就是呼叫 Script 的目錄。如果已存在相同名稱的定義檔，Script 會產生錯誤。

`.dic` 在企業搜尋中的大小上限是 8 MB。

程序

若要建立企業搜尋的停用字定義檔：

1. 以企業搜尋管理者的身分登入索引伺服器。 此使用者 ID 是在安裝 WebSphere II OmniFind Edition 時指定。
2. 輸入下列命令，其中 *XML_file* 是含有停用字清單的 XML 檔完整路徑，而 *DIC_file* 是停用字定義檔的完整路徑。

AIX、Linux 或 Solaris：`esstopworddictbuilder.sh` *`XML_file DIC_file`*
Windows: `esstopworddictbuilder.bat` *`XML_file DIC_file`*

建立停用字定義檔後，請使用企業搜尋管理主控台將定義檔新增至企業搜尋系統，並建立它與一或多個集合的關聯性。

只有產生的 .dic 檔案，才會上載到企業搜尋系統。請確定將原始的 XML 檔放在實施存取控制的環境中，並且確定您正常地備份檔案。您需要此 XML 檔以更新停用字定義檔。

自訂 Boost 字定義檔

您可以定義特定的字詞或多字字詞，以提高或降低出現該字詞的文件排序值。

Boost 定義檔中的每一個詞彙都與 Boost 因數相關聯，範圍從 -10 到 +10。您特別想在結果文件中看到的詞彙會配置較高的 Boost 因數，而那些完全不想讓它出現或與較高優先詞彙合併的那些詞彙，則會有較低的指定值。值 -1、0 及 1 沒有任何增值效果。

如果列示在 Boost 定義檔且具有特定 Boost 因數的查詢字詞出現在擷取的文件中，則文件排序值會視 Boost 值而升高或下降。指定給詞彙的 Boost 值是相對性的，因爲它也會受其他因數影響。因此，如果字詞 X 是以 B1 啓動且字詞 Y 是以 B2 啓動，而 B1 > B2，則 boost(X) >= boost(Y)。

Boost 字通常含有多字詞彙，如 *WebSphere Information Integrator OmniFind* 之類的產品名稱。 Boost 字定義檔所含的多字詞彙可在使用者查詢中正確地識別，而不需要以引號括住。

Boost 字定義檔與語言無關。

德文中的複合字詞可以在查詢中正確識別。複合字詞是當作單一字使用之兩個或多個字的組合。詞彙化複合，如 *Reisebüro* (旅行社) 不會被視爲複合。

會分解查詢中的複合字詞爲組成複合的個別字詞。如果組成複合字詞的個別字詞具有 Boost 值，則即使在字詞獨立地出現在文件中 (而不是作爲複合字詞的一部分) 時，指定的值會低於該字詞的 Boost 值，還是會排序擷取的文件。這會擴大搜尋範圍，而在只找到少數文件含有完整複合字詞的情況下，這是非常有用的。

例如，查詢字詞 `Versicherungspolice` (保險原則) 會傳回含有複合字詞 *Lebensversicherungspolice* (人壽保險原則) 及 *Haftpflichtversicherungspolice* (第三方保險原則) 的文件。如果 Boost 字定義檔含有單字 *Police* (原則)，則含有複合查詢字詞 `Versicherungspolice` 的文件會有指定的 Boost 值。

您必須在 XML 檔中列出詞彙及其 Boost 值，然後將該檔案轉換成 Boost 字定義檔，才能將它新增至企業搜尋系統。

您可以在企業搜尋管理主控台上選取要使用哪一個 Boost 字定義檔。每一個集合可以選取一個 Boost 字定義檔。而一個 Boost 字定義檔可由數個集合共用。

建立 Boost 字的 XML 檔

若要提高或降低某些結果文件的重要性，您必須在 XML 檔中指定哪些單字會影響文件排序。

關於本作業

列出 Boost 字的 XML 檔必須符合 XML 檔中指定的特定綱目。以下是 Boost 字的 XML 檔範例：

```
<?xml version="1.0" encoding="UTF-8"?>
<boostTerms xmlns="http://www.ibm.com/of/83/boostbuilder/xml">
    <!-- group boost terms by boost value-->
   <boostTermList boost="5">
    <!-- each term can specify the synonym expansion separatly-->
     <term useVariants="true">OmniFind Edition</term>
     <term useVariants="false">Edition</term>
<term>OmniFind</term>
   </boostTermList>
   <boostTermList boost="8">
        <term useVariants="true">WAS</term>
        <term>term9</term>
    </boostTermList>
</boostTerms>
```

限制

您可以將共用相同 Boost 值的字詞分組在 <boostTermList> 元素中，然而 Boost 值可以發生多次，例如，如果您要在 XML 檔中依字母順序排序 Boost 字。

Boost 字可以包含空格字元，但不能含有標點字元，如逗點 (,) 或垂直線 (|)，因爲這些字元可能會影響企業搜尋查詢語法。

Boost 字詞通常具有變體，如字首語或縮寫。您可以在 Boost 字定義檔中列舉所有變體；但是，如果您計劃使用同義字定義檔及 Boost 字定義檔，且已在同義字定義檔中新增詞彙及其變體，則不需要將這些變體也加入 Boost 字清單。而只需要針對新增至 Boost 字定義檔的變體，將屬性 useVariants 設爲 true 即可。如果這個詞彙列示在同義字定義檔中，則它在任何擷取文件中所發生的所有變體都會影響指定給這些文件的排序值。

您不需要列舉詞彙的正規化，例如移除重音或曲音 (企業搜尋會自動處理正規化)。比方說，如果您要將詞彙 météo 併入爲 Boost 字，則不需要同時加入詞彙 METEO。

程序

若要建立企業搜尋的 Boost 字清單：

1. 建立 XML 檔。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。XML 檔的 XSD 綱目稱爲 boostTerms.xsd，且內含於企業搜尋安裝的 *ES_INSTALL_ROOT*/packages/uima/configuration_xsd/ 中。
2. 在 <boostTerms xmlns="http://www.ibm.com/of/83/boostbuilder/xml"> 元素中併入對映。名稱空間 (在 xmlns 屬性中指定) 需要如所示範例一模一樣。
3. 新增 <boostTermList> 元素，以分組所有共用指定 Boost 值的字詞。

   Boost 值的範圍可以從 -10 到 10。例如，<boostTermList boost="-5"> 或 <boostTermList boost="5">。

   含有指定詞彙的文件重要性會依指定的 Boost 值而提高或降低。
4. 針對使用指定 Boost 值的每一個字詞，新增 <term> 元素。

   如果要併入列示在同義字定義檔中的 Boost 字變體，請將 <term> 元素的 useVariants 屬性設爲 true。預設值是 false。如果在同義字定義檔中找不到任何變體，則不會產生任何錯誤訊息。
5. 請重複之前的步驟，直到已指定使用者搜尋企業搜尋集合時要當成 Boost 字使用的所有詞彙爲止。

6. 儲存並結束 XML 檔。

建立 XML 檔後，您必須將它轉換成 Boost 字定義檔，然後才能新增至企業搜尋系統。

建立 Boost 字定義檔

在 XML 檔中建立或更新 Boost 字清單後，您必須將 XML 檔轉換成 Boost 字定義檔。

關於本作業

若要建立 Boost 字定義檔，請使用 WebSphere II OmniFind Edition 隨附的指令行工具，名爲 esboostworddictbuilder。這個工具是在 ES_INSTALL_ROOT/bin 目錄中。

工具的輸入是列出 Boost 字的 XML 檔，而工具的輸出就是 Boost 字定義檔。定義檔必須具有字尾 .dic。例如，c:\mydictionaries\productboostwords.dic。

這兩個檔案的預設位置，就是呼叫 Script 的目錄。如果已存在相同名稱的定義檔，Script 會產生錯誤。

.dic 在企業搜尋中的大小上限是 8MB。

程序

若要建立企業搜尋的 Boost 字定義檔：

1. 以企業搜尋管理者的身分登入索引伺服器。 此使用者 ID 是在安裝 WebSphere II OmniFind Edition 時指定。
2. 輸入下列命令，其中 *XML_file* 是含有 Boost 字清單的 XML 檔完整路徑，而 *DIC_file* 是 Boost 字定義檔的完整路徑。如果您也想使用同義字定義檔，請在 Boost 定義檔名稱後加上同義字定義檔的完整路徑。同義字定義檔的命名是選用的。

   UNIX®：esboostworddictbuilder.sh *XML_file DIC_file SYNDIC_file*
   Windows：esboostworddictbuilder.bat *XML_file DIC_file SYNDIC_file*

建立 Boost 字定義檔後，請使用企業搜尋管理主控台將定義檔新增至企業搜尋系統，並建立它與一或多個集合的關聯性。

只有產生的 .dic 檔案，才會上載到企業搜尋系統。請確定將原始的 XML 檔放在實施存取控制的環境中，並且具有適當的備份措施。您需要這個 XML 檔來更新 Boost 字定義檔。

相關工作

第 54 頁的『建立同義字定義檔』
在 XML 檔中建立或更新同義字清單之後，您必須將 XML 檔轉換爲二進位同義字定義檔。

企業搜尋中包括的文字分析

企業搜尋所含的文字分析包括文件語言偵測及斷詞法。

處理文件時，企業搜尋會判定該文件的語言，然後將輸入文字的串流分成不同的記號單元。

在搜尋期間，使用者或應用程式必須手動選取查詢語言。查詢字串會在索引中分斷、分析及搜尋。

文件及查詢字串分析可以分成：

- 基本非定義檔型支援。這包括空格及 n-gram 斷詞法。基本非定義檔型支援還包含句子斷詞法。
- 定義檔型語言支援。這包括單字和句子斷詞法，以及詞形還原。

  語言處理涉及詞彙分析，也就是建立輸入文字替代表示法的程序，該輸入文字會將所有可用的定義檔資料關聯至輸入文字中所辨識的記號。使用進階語言處理程序，可以大幅加強搜尋品質。

  相關概念

  『語言識別』
  企業搜尋必須先決定原始文件的語言，然後才能發生單字和句子斷詞法、字元正常化或詞形還原。

  第 66 頁的『非定義檔型斷詞法的語言支援』
  如果詞彙分析技術不支援文件的語言，則企業搜尋會提供 Unicode 型空格及 n-gram 斷詞法形式的基本支援。

語言識別

企業搜尋必須先決定原始文件的語言，然後才能發生單字和句子斷詞法、字元正常化或詞形還原。

企業搜尋可以自動偵測下列語言：

表 10. 自動語言識別的支援語言

南非荷蘭文	阿拉伯文	巴里文
巴斯克文	加泰蘭文	中文 (繁體及簡體)
捷克文	丹麥文	荷蘭文
英文	芬蘭文	法文
德文	希臘文	希伯萊文
冰島文	愛爾蘭文 (蓋爾文)	義大利文
日文	韓文	馬來文
挪威文 (巴克摩)	波蘭文	葡萄牙文
羅馬尼亞文	俄文	西班牙文
瑞典文	塔加洛文	泰文

表 10. 自動語言識別的支援語言 (繼續)

土耳其文	越南文	

企業搜尋中的語言處理程序會在索引期間 (而非查詢處理程序期間) 偵測原始文件的語言。

在企業搜尋中，您可以指定自動偵測文件的語言或選取要使用的語言。

如果選取自動語言偵測，而剖析器無法判定文件的語言，則剖析器會使用您在企業搜尋管理主控台建立搜索器時指定的語言。

如果您沒有選取自動語言偵測，則一律使用您指定的語言。 您可以藉由在企業搜尋管理主控台上編輯搜索器屬性，來指定文件語言。預設語言是英文。

您可以使用基本語言獨立技術 (如空格斷詞法及 n-gram 斷詞法)，處理沒有語言專用定義檔的文件。

企業搜尋語言偵測技術最適合用於單語文件。如果文件使用多語言，則會嘗試判定文件中所使用的最主要語言。然而，分析結果不一定是讓人滿意的。

文件的語言可以用來將您的搜尋結果限制於特定語言的文件。比方說，如果您在多語言文件集合中搜尋有關 Jacques Chirac 的文件，則可以限制只在搜尋結果中併入以法文撰寫的文件。您可以在企業搜尋管理主控台上選取進階搜尋選項，設定輸出文件的語言。

相關概念

第 65 頁的『企業搜尋中包括的文字分析』
企業搜尋所含的文字分析包括文件語言偵測及斷詞法。

『非定義檔型斷詞法的語言支援』
如果詞彙分析技術不支援文件的語言，則企業搜尋會提供 Unicode 型空格及 n-gram 斷詞法形式的基本支援。

非定義檔型斷詞法的語言支援

如果詞彙分析技術不支援文件的語言，則企業搜尋會提供 Unicode 型空格及 n-gram 斷詞法形式的基本支援。

Unicode 型空格斷詞法
這個語言處理方法在單字之間使用空格 (或空白空間) 作爲單字區隔字元。

N-gram 斷詞法
這個語言處理方法將 *n* 個字元的重疊順序視爲單一個字。在許多擷取作業中，這個簡單的斷詞法方法就已經夠用了。

這些方法與任何語言定義檔無關，且不涉及更準確的語言處理技術，如基礎詞形還原。

N-gram 斷詞法適用於不使用空格作爲區隔字元的語言，如泰文。相同的方法也適用於希伯萊文及阿拉伯文。雖然這兩種語言都使用空格區隔字元，但 n-gram 斷詞法傳回的結果會比 Unicode 型空格斷詞法的基本形更好。

當您建立集合時，還可以使用 n-gram 斷詞法，選擇性地選取符記化中文及日文文件。

若要在 n-gram 斷詞法期間移除所有空格字元 (例如，換行或定位字元)，則必須在開始剖析文件之前，先開啓 *ES_NODE_ROOT*/master_config/<CollectionID>.parserdriver 中稱爲 collection.properties 之檔案內的參數設定。移除空格字元所需的參數包括：

- **removeCjNewLineChars**：如果設爲 *true*，則此參數會移除在中文或日文字元之間以任何順序出現的換行字元及定位字元。預設值是 `removeCjNewlineChars=false`。
- **removeCjNewLineCharsMode**：如果設爲 *all*，則無論字元環境定義爲何，此參數都會移除空格字元。例如，還會移除英文文字中的空格字元。如果您要使用此選項，則必須將參數新增至性質檔。只有 `removeCjNewlineCharsMode=all` 是有效的，所有其他值都會被忽略。

相關概念

第 65 頁的『企業搜尋中包括的文字分析』
企業搜尋所含的文字分析包括文件語言偵測及斷詞法。

第 65 頁的『語言識別』
企業搜尋必須先決定原始文件的語言，然後才能發生單字和句子斷詞法、字元正常化或詞形還原。

符記化數字字元為 n-gram 記號

若要將雙位元組字元之外的數字字元符記化爲 n-gram 記號，您必須在空格及 n-gram 記號器描述子檔案中開啓參數設定。

關於本作業

空格及 n-gram 記號器中數字字元的預設處理方式是將所有數字字元視爲空格分段的記號。若要將數字字元符記化爲 n-gram 記號，您必須變更註解程式描述子檔案中的 n-gram 模式設定。您無法使用企業搜尋管理主控台來變更此設定。

程序

預設 n-gram 模式設定稱爲正常，並將數字字元及 SBCS 字元視爲以空格分段的字元。若要啓用數字 n-gram 模式，請：

1. 停止集合的剖析器。
2. 停止集合的執行時期。
3. 開啓 *ES_NODE_ROOT*/master_config/<CollectionID>.parserdriver/specifiers 目錄中稱爲 jtok.xml 的註解程式描述子檔案。 CollectionID 是在建立集合時針對集合指定 (或由系統指派) 的 ID。
4. 將 **NgramMode** 參數設定從正常變更爲數字。
5. 重新啓動集合的剖析器。
6. 重新啓動執行時期。

定義檔型斷詞法的語言支援

如果正確地偵測到文件的語言且可以使用語言專用定義檔，則會套用適當的語言處理程序。

斷詞法是將輸入文件分成不同詞元的程序。此程序包括下列部分語言處理活動：

斷詞 (Word segmentation)
斷詞用於不在單字之間使用空格 (或區隔字元) 的語言，如日文和中文。

詞形還原 (Lemmatization)

詞形還原是一種語言處理形式，決定文字中每一個字形的詞形。單字的*詞形*包含了基礎詞形，並加上共用相同詞類部分的變化形。例如，*go* 的詞形包含 *go*、*goes*、*went*、*gone* 及 *going*。名詞的詞形分成單數和複數形 (如 *calf* 及 *calves*)。形容詞的詞形分成比較級和最高級形式 (如 *good*、*better* 及 *best*)。代名詞的詞形分成相同名詞的不同格 (如 *I*、*me*、*my* 及 *mine*)。

詞形還原需要檢索和搜尋的定義檔。

企業搜尋會索引詞形和變化形，並還原查詢中所有變化形。詞形還原會尋找含有查詢中變化形變體的文件，以加強搜尋品質。例如，當查詢含有單字 *mouse* 時，會尋找含有單字 *mice* 的文件。

縮寫分割 (Contraction splitting)

識別縮詞並將它們分割成元件組件，以增進搜尋品質。例如：

wouldn't 分割成 *would + not*
Horse's 分割成 *Horse + 's*

附著語素識別

附著語素是特殊的縮詞形式，可判定元件組件來增進搜尋品質。*附著語素*是作用類似詞綴及單字的一種元素。然而，附著語素很難識別，因爲它們也是字形的一部分。與其他語素 (單字結構) 現象不同，附著語素發生在語法結構中，而它們與單字的連接並不是字形規則的一部分。例如：

reparti-lo-emos 分成元件 *repartir + lo + emos*
l'avenue 分成元件 *le + avenue*
dell'arte 分成元件 *dello + arte.*

非字母型字元識別

語言處理可辨識非字母型字元。依據內部語言相關邏輯，部分非字母型字元會當成不同類型的個別詞元傳回，而部分則會加以分組。

例如，撇號在附著語素中會被視爲單字組件，而在不明縮寫的情況下，則會被視爲句點 (句號)。URL、電子郵件位址及日期會分割成數個記號。

縮寫識別 (Abbreviation recognition)

語言處理會將定義檔中的縮寫辨識爲一個詞元。如果縮寫不在定義檔中，則縮寫會被辨識爲詞彙項目，但該縮寫將不會有任何相關的定義檔資訊。

正確地辨識縮寫是句子識別的重點。例如，縮寫結尾的句點不一定是句子的結尾。

句尾標記識別 (End-of-sentence marker recognition)

語言處理可正確地識別句子斷詞法的句尾標記。

定義檔型語言支援適用於下列語言：

表 11. 支援語言

阿拉伯文	義大利文
中文 (簡體及繁體)	日文
捷克文	韓文
丹麥文	挪威文 (巴克摩)
荷蘭文	波蘭文

表 11. 支援語言 (繼續)

英文	葡萄牙文 (國家及巴西)
芬蘭文	俄文
法文 (國家及加拿大)	西班牙文
德文 (國家及瑞士)	瑞典文
希臘文	

相關概念

『日文的斷詞』
如果要以日文辨識純文字文件或查詢字串，則企業搜尋會利用最適用於日文的形態分析技術， 執行相關的斷詞。

『日文的正體字變體』
日文使用許多正體字變體。Katakana 變體是最重要的，因爲 Katakana 通常是用於外國字的拼寫和發音。 日文經常使用許多 Katakana 變體。

日文的斷詞

如果要以日文辨識純文字文件或查詢字串，則企業搜尋會利用最適用於日文的形態分析技術，執行相關的斷詞。

這個最佳化的範例是單字分解。日文使用大量的複合字。這些字會分解成最佳大小的記號，以達到較好的搜尋結果。同時也會分解變化形和前置詞以增進搜尋效能。

相關概念

第 67 頁的『定義檔型斷詞法的語言支援』
如果正確地偵測到文件的語言且可以使用語言專用定義檔，則會套用適當的語言處理程序。

『日文的正體字變體』
日文使用許多正體字變體。Katakana 變體是最重要的，因爲 Katakana 通常是用於外國字的拼寫和發音。 日文經常使用許多 Katakana 變體。

日文的正體字變體

日文使用許多正體字變體。Katakana 變體是最重要的，因爲 Katakana 通常是用於外國字的拼寫和發音。日文經常使用許多 Katakana 變體。

企業搜尋使用變體定義檔將一般的 Katakana 變體對映至其基本形式 (類似詞形)，以便能找到所有文件，包括含有查詢字串中 Katakana 單字之正體字變體的文件。

企業搜尋也支援一般的 Okurigana 變體，這是以平假名結尾的漢字單字。

相關概念

第 67 頁的『定義檔型斷詞法的語言支援』
如果正確地偵測到文件的語言且可以使用語言專用定義檔，則會套用適當的語言處理程序。

『日文的斷詞』
如果要以日文辨識純文字文件或查詢字串，則企業搜尋會利用最適用於日文的形態分析技術， 執行相關的斷詞。

停用字移除

在企業搜尋中，會從多字查詢中移除所有停用字 (例如， *a* 及 *the* 之類的通用字)，以增加搜尋效能。

日文的停用字識別以文法資訊爲基礎，例如，企業搜尋會辨識單字是名詞還是動詞。針對其他語言，企業搜尋使用特殊的清單。

在下列情況下，不會在查詢處理作業期間移除任何停用字：

- 查詢中的所有單字都是停用字。如果在停用字處理作業期間移除所有查詢字詞，則結果集是空的。若要確保會傳回搜尋結果，請在查詢字詞全部是停用字的情況下，停止使用停用字移除。例如，如果*車* 一字是停用字，且您搜尋*車*，則搜尋結果會包含符合*車* 一字的文件。如果您搜尋*別克車*，則搜尋結果僅會包含符合*別克*一字的文件。
- 查詢中的單字之前是加號 (+)。
- 單字是完全相符項的一部份。
- 單字在詞組內部，例如，「我愛我的車」。

相關概念

『字元正常化』
字元正常化是可以增進索引率的一種程序。使用字元正常化增進索引率， 表示即使文件不完全符合查詢，還是可以擷取更多文件。

字元正常化

字元正常化是可以增進索引率的一種程序。使用字元正常化增進索引率，表示即使文件不完全符合查詢，還是可以擷取更多文件。

企業搜尋使用 Unicode 相容性正常化，其中包括亞洲半形至全形字元的正常化。

企業搜尋也可移除 Katakana 中間點 (當成日文的複合字區隔字元使用)。

字元正常化的其他形式包括：

大小寫正常化
例如，搜尋 *usa* 時會尋找含有 *USA* 的文件。

曲音符號擴充
例如，搜尋 *schön* 時會尋找含有 *schoen* 的文件。

重音符號移除
例如，搜尋 *e* 時會尋找含有 *é* 的文件。

其他區別發音符號移除
例如，搜尋 *c* 時會尋找含有 *ç* 的文件。

連字擴充
例如，搜尋 *ae* 時會尋找含有 *Æ* 的文件。

所有正常化都適用於兩種方式。您可以在搜尋 *USA* 時尋找含有 *usa* 的文件，在搜尋 *é* 時尋找單字中有 *e* 的文件等等。這些正常化也可以合併。例如，搜尋 *METEO* 時尋找含有 *météo* 的文件。

正常化是以 Unicode 字元內容爲基礎，和語言無關。例如，企業搜尋支援希伯萊文的區別發音符號移除，以及阿拉伯文的連字擴充。

相關概念

第 70 頁的『停用字移除』
在企業搜尋中，會從多字查詢中移除所有停用字 (例如， *a* 及 *the* 之類的通用字)，以增加搜尋效能。

正規表示式註解程式

正規表示式註解程式可讓您執行自訂文字分析，而無需實作您自己的文字分析引擎。根據您可以自行定義的一組規則 (正規表示式)，正規表示式註解程式會偵測純文字文件中的資訊結構，並在共用分析結構中建立已偵測之資訊的註解。

正規表示式註解程式會根據正規表示式，偵測純文字文件中資訊的實體或單位，例如，電話號碼，產品型號、建築物及房間的號碼或地址。如果其中一項正規表示式符合部分文件文字，則正規表示式註解程式會建立相對應的註解程式，來涵蓋相符的部分資訊。這些註解儲存在共用分析結構中，並可以稍後藉由將這些分析結果對映至企業搜尋索引，來使用共用分析結構到索引的對映檔。此外，可以建立共用分析結構到資料庫的對映檔，以在可支援 JDBC 的資料庫中儲存註解。

您定義的規則集 (正規表示式) 儲存在 XML 配置檔 (也稱爲規則集檔案) 中。正規表示式註解程式包含處理這些正規表示式的分析邏輯。它支援 Java 1.4 中的正規表示式語法。

正規表示式註解程式的類型系統說明，必須定義由正規表示式註解程式所建立並使用的註解類型及特性。根據正規表示式註解程式之應用程式區域的複雜性 (例如，如果所需的類型多於在提供之正規表示式註解程式中定義的類型)，其他輸入及輸出功能必須定義在正規表示式註解程式描述子中。描述子中使用的類型必須符合註解程式類型系統說明中的類型。

正規表示式註解程式內含在企業搜尋中作爲可部署的 PEAR (處理程序引擎保存檔) 檔，使用範例規則配置該檔案，可以偵測電話號碼、URL 及電子郵件位址。

相關概念

第 76 頁的『規則集檔案』
在正規表示式註解程式中，XML 規則集檔案會以正規表示式形式來定義用於剖析純文字文件的規則。

相關工作

第 76 頁的『定義正規表示式規則』
規則集定義符合文件中文字的正規表示式，以及當型樣相符時正規表示式註解程式必須採取的動作。

相關參考

第 80 頁的『註解程式描述子』
正規表示式註解程式 XML 描述子包含執行註解程式所需之正規表示式註解程式的相關敘述性資訊。

第 83 頁的『日誌記載』
所有正規表示式註解程式的日誌訊息都會寫入現行集合的日誌檔中。

使用正規表示式註解程式的簡易語意搜尋

企業搜尋包括使用一組規則所預先配置的正規表示式分析引擎，您可以啓用它來偵測純文字文件中的電話號碼、URL 及電子郵件位址。

您可以使用此正規表示式分析引擎的配置範例，來啓用企業搜尋，以在文件中尋找實際的電話號碼，而無需在文件中尋找關鍵字*電話號碼*。若要查詢由正規表示式註解程式所偵測的建構，還要提供共用分析結構到索引之對映檔的範例。此外，還會示範一種簡易方法，讓您透過簡單的關鍵字發出有效的語意查詢。此方法使用企業搜尋同義字支援，自動將簡易關鍵字查詢擴充爲語意查詢。還會提供說明此機製的範例同義字定義檔。您可以在 ES_INSTALL_ROOT/packages/uima/regex 找到要搭配使用正規表示式註解程式與配置範例所需的所有檔案。

對於許多應用程式實務而言，只要稍微修改搭配配置範例的正規表示式規則，將正規表示式註解程式調整爲符合您的需求可能就足夠了。

然而，若要完全自訂註解程式，則建議您使用 UIMA SDK。爲達此目的，正規表示式註解程式也要併入位於 ES_INSTALL_ROOT/packages/uima/ 的企業搜尋基礎註解程式套件中。

相關工作

『啓用使用正規表示式註解程式的簡易語意搜尋』
若要使用同義字啓用簡易語意搜尋，您必須將正規表示式註解程式、共用分析結構到索引的對映檔及範例同義字定義檔新增至企業搜尋系統，並建立這些資源與集合的關聯性。

第 79 頁的『自訂正規表示式註解程式』
您可以自訂正規表示式註解程式的配置範例來偵測新的實體 (例如，產品序號)，或稍微變更範例規則集與類型系統檔案，使正規表示式規則適合現有實體 (例如，偵測公司專用電話號碼)。

第 11 頁的『檢視基本註解程式及自訂文字分析結果』
若要檢視在剖析之後由企業搜尋中任何註解程式所產生的分析結果，您必須更新文件集合屬性，以產生儲存在共用分析結構中之分析結果的可讀 XML 版本。

啓用使用正規表示式註解程式的簡易語意搜尋

若要使用同義字啓用簡易語意搜尋，您必須將正規表示式註解程式、共用分析結構到索引的對映檔及範例同義字定義檔新增至企業搜尋系統，並建立這些資源與集合的關聯性。

此後，正規表示式註解程式會在剖析階段期間處理文件，索引器會將自訂分析結果新增至索引，搜尋服務可以利用提供的語意同義字定義檔，透過自動擴充至語意查詢的簡易關鍵字搜尋自訂分析結果。

程序

若要啓用簡易語意搜尋，請：

1. 使用企業搜尋管理主控台，將名爲 of_regex.pear 且位於 *ES_INSTALL_ROOT*/packages/uima/regex 的正規表示式自訂文字分析引擎新增至企業搜尋系統。
2. 建立正規表示式文字分析引擎與集合的關聯性。
3. 將共用分析結構新增至 *ES_INSTALL_ROOT*/packages/uima/regex 目錄中稱爲 of_sample_regex_cas2index.xml 的索引對映檔中。這會將正規表示式註解程式產生的自訂分析結果 (註解) 對映至企業搜尋索引中的可搜尋跨距。然後，您可以使用 XML 片段 或 XPath 查詢來搜尋這些跨距。

4. 搜索、剖析並建立集合索引。 此時，完成索引之後，您可以使用 XML 片段表示式來輸入 XML 搜索查詢，例如，使用搜尋應用程式輸入 `@xmlf2::'<#phonenumber>'`。然而，使用同義字啓用語意搜尋的目的，是讓您使用類似於 `Barbara 電話號碼`的查詢，並讓系統將查詢轉換爲 Barbara @xmlf2::'<#phonenumber>'。
5. 使用管理主控台，將所提供的範例二進位同義字定義檔 (位於 *ES_INSTALL_ROOT*/packages/uima/regex 目錄中，名稱爲 of_sample_synonym_dic.dic) 新增至企業搜尋系統。 您可以變更來源 XML 範例定義檔，或者使用其作爲基準來建立您自己的定義檔，然後，使用 essyndictbuilder 工具將建立的定義檔轉換爲新定義檔。XML 範例同義字定義檔名爲 of_sample_synonym_dic.xml，也位於 *ES_INSTALL_ROOT*/packages/uima/regex。
6. 建立同義字定義檔與集合的關聯性，並啓動 (或重新啓動) 集合的搜尋服務。
7. 在搜尋應用程式中，選取選項，以使用擴充的語意自動搜尋同義字。啓用此選項之後，搜尋應用程式會重寫 XML 片段查詢的基礎關鍵字查詢，並併入用於尋找可以識別電話號碼、電子郵件位址及 URL 之可搜尋跨距的表示式。
8. 在搜尋應用程式中，輸入要求電話號碼的查詢，例如，`barbara 電話號碼`。 查詢會搜尋含有三個關鍵字 (*barbara*、*電話*、*號碼*) 的文件，以及含有關鍵字 *barbara* 且其中的數字及字元跨距均與針對電話號碼所定義之正規表示式相符的文件。在搜尋結果中，會強調顯示找到的關鍵字及電話號碼。

您可以在提供的範例同義字定義檔中查看哪些關鍵字會轉換爲語意查詢。

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups xmlns="http://www.ibm.com/of/822/synonym/xml">
  <synonymgroup>
<synonym>telephone number</synonym>
<synonym>phone number</synonym>
<synonym>telephone nbr</synonym>
<synonym>phone nbr</synonym>
<synonym>@xmlf2::'&lt;#phonenumber/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
<synonym>facsimile number</synonym>
<synonym>fax number</synonym>
<synonym>facsimile nbr</synonym>
<synonym>fax nbr</synonym>
<synonym>@xmlf2::'&lt;#phonenumber/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
<synonym>e-mail address</synonym>
<synonym>email address</synonym>
<synonym>@xmlf2::'&lt;#email/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
<synonym>URL</synonym>
<synonym>unified resource locator</synonym>
<synonym>Web address</synonym>
<synonym>@xmlf2::'&lt;#url/&gt;'</synonym>
  </synonymgroup>
</synonymgroups>
```

相關概念

第 73 頁的『使用正規表示式註解程式的簡易語意搜尋』
企業搜尋包括使用一組規則所預先配置的正規表示式分析引擎，您可以啓用它來偵測純文字文件中的電話號碼、URL 及電子郵件位址。

規則集檔案

在正規表示式註解程式中，XML 規則集檔案會以正規表示式形式來定義用於剖析純文字文件的規則。

規則會循序指定文件文字中的位置、註解程式必須尋找的相符項，以及在找到相符項之後要採取的動作。

已呼叫正規表示式註解程式時，會編譯包含正規表示式型樣的 XML 規則集檔案，並將其與部分文件文字相比對。如果找到相符或部分相符項，則會建立與特定規則相關聯的註解，並儲存在共用分析結構中。

規則中使用的類型必須定義在正規表示式註解程式的類型系統說明中。

正規表示式註解程式從 XML 規則集檔案中的第一個規則開始，一次處理一個規則。針對每一個規則，將相對應的已編譯正規表示式與之前步驟中建立的註解相比對，例如，在正規表示式註解程式之前處理文件的註解程式所建立的註解。符合規則的註解必須與在正規表示式註解程式描述子中指定之輸入功能的類型相同。

如果找到相符項，所激發規則中建立的註解類型也必須在正規表示式註解程式描述子中指定爲有效的輸出功能類型。之前規則建立的新註解，可用作稍後在 XML 規則集中激發之規則的輸入註解。

相關概念

第 73 頁的『正規表示式註解程式』
正規表示式註解程式可讓您執行自訂文字分析，而無需實作您自己的文字分析引擎。根據您可以自行定義的一組規則 (正規表示式)，正規表示式註解程式會偵測純文字文件中的資訊結構，並在共用分析結構中建立已偵測之資訊的註解。

相關工作

『定義正規表示式規則』
規則集定義符合文件中文字的正規表示式，以及當型樣相符時正規表示式註解程式必須採取的動作。

相關參考

第 80 頁的『註解程式描述子』
正規表示式註解程式 XML 描述子包含執行註解程式所需之正規表示式註解程式的相關敘述性資訊。

第 83 頁的『日誌記載』
所有正規表示式註解程式的日誌訊息都會寫入現行集合的日誌檔中。

定義正規表示式規則

規則集定義符合文件中文字的正規表示式，以及當型樣相符時正規表示式註解程式必須採取的動作。

關於本作業

XML 規則集檔案必須遵循下列範例中概述的規則語法。這是範例正規表示式註解程式的規則集檔案，可辨識電話號碼、URL 及電子郵件位址。

最上層元素是 <ruleSet> 元素，其由一或多個 <rule> 元素組成。每一個 <rule> 元素依次定義由屬性 regEx 以及 matchStrategy 和 matchType 屬性組成的 Java 正規表示式。在指定註解 ID 及註解類型的 <createAnnotation> 元素中定義該動作。

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ruleSet.xsd">
<!-- Phone Number -->
<!-- This rule matches different ways of writing telephone numbers,
for example, 01234-12345, 01234 / 122-32, (001234)12345,
+49 (0) 123412345, (123) 123 1234,
1-800-IBM-4YOU -->
<rule regEx="(?x)(\s|\b)(
0{1,2}[1-9]{1}[0-9]{1,5}\x20?[-/\\]\x20?[1-9]{1}([0-9]{1,8}-?)
  {1,3}[0-9]{1,}
|\(0[1-9]{1}[0-9]{1,3}\)\x20?[1-9]{1}[0-9]{2,8}
|\(00[1-9]{1}[0-9]{1,8}\)\x20?[1-9]{1}[0-9]{2,10}
|\((0\x20?[1-9]{1}[0-9]{1,3}|00\x20?[1-9]{1}[0-9]{1,8})\)\x20?[1-9]
{1}[0-9] {1,3}(\x20[0-9]{2,4}){1,5}
|(0\x20?[1-9]{1}[0-9]{1,3}|00\x20?[1-9]{1}[0-9]{1,8})\x20?[/\\]\x20?
[1-9]{1} [0-9]{1,3}(\x20[0-9]{2,4}){1,5}
|\(?\+[1-9]{1}[0-9]{0,3}\)?([-\x20]|\x20?\(0\))[-\x20]?[1-9]
   {1}[0-9]{1,10}
|\(?\+[1-9]{1}[0-9]{0,3}\)?([-\x20]|\x20?\(0\))[-\x20]?[1-9]
{1}[0-9]{1,3}[-\x20]([0-9]{2,5}[-\x20]?){1,4}
|(1-)?[0-9]{3}-[0-9]{3}-[0-9]{4}
|\([1-9]{1}[0-9]{2}\)\x20[0-9]{3}[-\x20][0-9]{4}
|1-(800|888|877|866)-([A-Z0-9]{7}|[A-Z0-9]{3}-[A-Z0-9]
   {4}|[A-Z0-9]{4}-[A-Z0-9]{3})
)(?!(\d|\x20\d|-\d))(\s|\b)"
matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
<createAnnotation id="phonenumber" type="com.ibm.es.uima.PhoneNumber">
<begin group="0"/>
<end group="0"/>
  </createAnnotation>
 </rule>
<!-- potential Phone Number -->
<!-- This rule matches numbers that resemble telephone numbers but could
also be anything else. For example, 0123 1234 123,
+123456789, 123 123 1234 -->
<rule regEx="(?x)(\s|\b)(
  0[1-9]{1}[0-9]{1,3}\x20[1-9]{1}[0-9]*\x20?([0-9]{2,}\x20?)+
  |00\x20?[1-9]{1}[0-9]{0,3}\x20[1-9]{1}[0-9]{1,3}\x20?[1-9]
{1}([0-9]{2,}\x20?)+
  |\+[1-9]{1}[0-9]{0,3}[1-9]{1}[0-9]{6,}
  |[1-9]{1}[0-9]{2}\x20[0-9]{3}\x20[0-9]{4}
  )(?!(\d|\x20\d|-\d))(\s|\b)"
matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
<createAnnotation id="potential_phonenumber"
type="com.ibm.es.uima.PotentialPhoneNumber">
<begin group="0"/>
<end group="0"/>
  </createAnnotation>
 </rule>
<!-- URL Annotation -->
<!-- This rule matches URLs, for example, http://www.ibm.com -->
<rule regEx="(?x)(\s|\b)(
    http://[\w\-]+([\.][\w\-]+)+([/][\w\~\(\)\-\?=%\u0026\#]*)*
    |www.[\w\-]+([\.][\w\-]+)+([/][\w\~\(\)\-\?=%\u0026\#]*)*
   )(\s|\b)"
matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
<createAnnotation id="url" type="com.ibm.es.uima.URL">
<begin group="0"/>
<end group="0"/>
  </createAnnotation>
 </rule>
<!-- Email Annotation -->
```

```
<!-- This rule matches e-mail addresses, for example, yourName@domain.com -->
<rule regEx="(?x)(\s|\b)(
    [a-zA-Z0-9][\w\.-]*[a-zA-Z0-9]@[a-zA-Z0-9]([\.-]?\w)*\.[a-zA-Z]
{2,3})(\s|\b)"
matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
<createAnnotation id="email" type="com.ibm.es.uima.Email">
<begin group="0"/>
<end group="0"/>
  </createAnnotation>
 </rule>
</ruleSet>
```

程序

若要針對定義自訂正規表示式的正規表示式註解程式建立 XML 規則集，請：

1. 建立 XML 檔。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。XML 規則集檔案的 XSD 綱目稱爲 ruleSet.xsd，其位於 *ES_INSTALL_ROOT*/packages/uima/regex/ 目錄的企業搜尋安裝中。
2. 在 `<ruleSet xmlns="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="ruleSet.xsd">` 元素中併入對映。名稱空間在 xmlns 屬性中指定，且必須與所示範例完全相同。
3. 新增包含 `regEx` 屬性 (包含正規表示式型樣)、`matchStrategy` 屬性及 `matchType` 屬性的 `<rule>` 元素。

   註解程式完全支援 Java 1.4 正規表示式語法。如需正規表示式及檢視完整語法的指示，請參閱位於 http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html 的 Java 文件。

   `matchStrategy` 指定搜尋方式，例如，是否必須找到文件中所有的相符項，或者文字相符項是否必須完全相符。您可以採用下列三種不同的相符策略：

   - `matchFirst` 在找到符合正規型樣的第一個文字序列時停止
   - `matchAll` 尋找文件中符合正規型樣的所有文字序列
   - `matchComplete` 僅完全相符的文字序列才算符合。例如，如果我們具有型樣“foo”，則只能符合字詞 ″foo″，相符項中不會產生 ″foobar″。

   `matchType` 決定規則要符合的註解類型。例如，您可以利用此方法將要符合的正規表示式限制在現有記號註解內。這會避免符合規則內太多的內容。可能的類型是註解程式容許的輸入註解類型 (定義在註解程式描述子中)，如 uima.tt.DocumentAnnotation、uima.tt.ParagraphAnnotation，及使用者定義的類型，如 foo.bar.MyAnnotation。有時，一個規則的輸出類型可以用作後續規則的輸入類型。`matchType` 可讓您限制特定規則的搜尋範圍。
4. 新增 `<createAnnotation>` 元素，當找到相符項時，此元素會定義正規表示式註解程式要採取的動作。

   每一個 `createAnnotation` 元素都具有兩個屬性：

   - `id` 唯一識別註解，且用於參照註解
   - `type` 指定已建立的註解類型
5. 新增下列元件元素，其定義 `<createAnnotation>` 元素的相符位置：
   - 必備元素：`<begin>` 指定相符項開始的位置。此元素有兩個屬性：

  - 必備元素：group 辨識 Java 擷取群組。可針對其指派 0 (完全文字序列相符) 至 9 (多個擷取群組) 之間的值
  - 選用元素：location 指出相符群組內的位置 (關於括弧的定位)，爲 start (左括弧) 或 end (右括弧)。
- 必備元素：<end> 指定相符項結束的位置。此元素有兩個屬性：
  - 必備元素：group 識別擷取群組。可針對其指派 0 (完全文字序列相符) 至 9 (後續及更小的相符群組) 之間的值
  - 選用元素：location 指出相符群組內的位置 (關於括弧的定位)，爲 start (左括弧) 或 end (右括弧)。
- 選用元素：<setFeature> 建立特性，並將其指派給註解。此元素有兩個屬性：
  - name 是特性名稱，如您在類型系統說明所定義
  - type 指定特性值的類型，爲 String、Integer、Float 及 Reference。類型必須與針對註解程式類型系統說明中特性所定義的範圍類型相同。

    類型 Reference 的特性可用來建立兩個註解之間的鏈結，以模型化語意關係。<setFeature> 元素內容必須設爲要建立鏈結之 <createAnnotation> 元素的 id。

相關概念

第 76 頁的『規則集檔案』
在正規表示式註解程式中，XML 規則集檔案會以正規表示式形式來定義用於剖析純文字文件的規則。

自訂正規表示式註解程式

您可以自訂正規表示式註解程式的配置範例來偵測新的實體 (例如，產品序號)，或稍微變更範例規則集與類型系統檔案，使正規表示式規則適合現有實體 (例如，偵測公司專用電話號碼)。

已修改的規則集檔案及類型系統說明必須新增至正規表示式處理作業引擎保存檔 (PEAR 檔)。更新 PEAR 檔之後，您可以再次將自訂的正規表示式文字分析引擎新增至企業搜尋系統。

若要更詳細地自訂正規表示式註解程式，強烈建議您使用 UIMA SDK 工具。這些工具可協助您建立或更新類型系統說明及描述子檔案、可能使註解程式相結合以形成聚集分析引擎，以及建立內含所有必要資源的新處理作業引擎保存檔 (PEAR 檔) 以使用企業搜尋中的註解程式。如需可支援您完成這些任務之工具的相關資訊，請參閱 UIMA SDK 文件。

程序

若要透過新增規則及實體來調整正規表示式註解程式，或變更現有規則，您可以更新已提供的範例正規表示式註解程式 PEAR 檔，如下所示：

1. 在系統中建立稱爲 xml 的新目錄。
2. 將 *ES_INSTALL_ROOT*/packages/uima/regex/ 目錄中的範例規則檔案 of_sample_regex_rules.xml 複製到 xml 目錄，然後修改該檔案，使其包括自訂型樣相符規則。 若要避免 XML 語法錯誤，請選擇使用 XML 編輯器或 XML 編寫工具。

3. 將相對應的類型系統說明檔案 of_sample_typesystem.xml 從目錄 ES_INSTALL_ROOT/packages/uima/regex/ 複製到 xml 目錄中，並修改該檔案，使其包括新規則所需之類型的定義。
4. 如果僅新增少數規則或變更現有規則，則不需要變更註解程式描述子。如果您計劃進行其他變更，或使用其他自訂分析步驟，請檢查是否必須修改註解程式描述子。
5. 使用您選擇的保存公用程式，以將正規表示式註解程式 PEAR 檔的複本更新爲包括您的兩個更新檔案。 例如，將 of_regex.pear 檔案從 ES_INSTALL_ROOT/packages/uima/regex/ 複製到您建立之 xml 目錄的母目錄。然後，使用 Java Jar 指令行工具 (例如，部分 IBM Java SDK) 從該母目錄發出下列指令：

   ```
   "jar -uf of_regex.pear -C xml/ of_sample_regex_rules.xml"
   "jar -uf of_regex.pear -C xml/ of_sample_regex_typesystem.xml"
   ```

6. 使用企業搜尋管理主控台，將作爲自訂文字分析引擎的正規表示式註解程式新增至企業搜尋系統，並建立其與測試文件集合的關聯性。
7. 透過更新文件集合內容來檢查由正規表示式註解程式所產生的分析結果，以產生分析結果的可讀 XML 輸出，該分析結果儲存在使用 XCAS 傾出特性的共用分析結構中。
8. 處理測試文件，並使用「XCAS 註解檢視器」來檢視 XML 檔的內容。
9. 如果您對註解程式根據自訂正規表示式所建立的註解很滿意，則再次編輯文件集合內容，以停用產生分析結果之可讀 XML 輸出的剖析器。如果需要進一步修改規則集檔案，則必須重複更新 PEAR 檔的步驟。
10. 建立共用分析結構到索引的對映檔以建立分析結果的索引，或者如果您要將結果新增至資料庫，則建立共用分析結構到資料庫的對映檔。 您可以使用已提供之共用分析結構到索引的對映檔範例作爲開始點，建立您的共用分析結構到索引的對映檔。
11. 使用企業搜尋管理主控台新增對映檔，並建立它們與完整文件集合的關聯性。
12. 使用 XML 片段或 XPath 查詢來搜尋註解，或者選擇性地使用擴充的語意來進行同義字搜尋。

相關概念

第 73 頁的『使用正規表示式註解程式的簡易語意搜尋』
企業搜尋包括使用一組規則所預先配置的正規表示式分析引擎，您可以啓用它來偵測純文字文件中的電話號碼、URL 及電子郵件位址。

相關工作

第 11 頁的『檢視基本註解程式及自訂文字分析結果』
若要檢視在剖析之後由企業搜尋中任何註解程式所產生的分析結果，您必須更新文件集合屬性，以產生儲存在共用分析結構中之分析結果的可讀 XML 版本。

註解程式描述子

正規表示式註解程式 XML 描述子包含執行註解程式所需之正規表示式註解程式的相關敘述性資訊。

在下列情況下，如果您只使用正規表示式註解程式，且沒有其他自訂分析步驟，則只需要變更描述子：

- 您要變更規則集檔案的檔名 (在 <externalResourceDependencies> 元素中)。

- 您要使用多個規則集檔案。
- 您要變更類型系統說明檔的名稱。

在下列情況下，如果您是使用其他自訂分析步驟，則需要變更描述子：

- 您希望自訂分析使用由正規表示式註解程式所建立的註解。在此情況下，您必須更新註解程式描述子中的輸出功能。
- 您已定義正規表示式規則，該規則必須符合在之前自訂分析步驟中建立的註解類型。在此情況下，您必須更新註解程式描述子中的輸入功能。

使用 UIMA SDK 工具，來建立或更新註解程式描述子及重建處理作業引擎保存檔 (.pear 檔)，該檔案包括要在企業搜尋中使用註解程式所需的全部資源。如需可在這些作業中支援之工具的相關資訊，請參閱 UIMA 文件。

配置參數

正規表示式註解程式只有一個配置參數，稱爲 String2NumberImpl，您必須將其設爲實作 com.ibm.uima.an_regex.String2Number 介面之類別的名稱。必須同時提供正規表示式註解程式與此類別的實作，否則將會發生異常狀況。如果您要自訂正規表示式註解程式以符合您的需要，則可以透過傳遞 XML 描述子檔案中的類別名稱，來提供您自己的 String2Number 介面實作。

String2Number 介面宣告了兩種方法，toInt(String) 及 toFloat(String)，其可以將整數或浮點值的字串表示法轉換爲相對應的整數或浮點值。這兩種方法用於將包含分隔字元的數字轉換爲有效的 Java「整數」或「浮點」值。

com.ibm.uima.an_regex.String2Number_impl 的預設實作會考量使用句點 (.) 作爲小數點符號，使用逗點 (,) 作爲千位分隔字元。例如，如果在純文字文件中找到 1,999.00，則 toInt 會將它轉換爲 1999。toFloat 會傳回 1999.00。

範例

描述子的配置參數區段如下所示：

```
<configurationParameters>
 <configurationParameter>
<name>String2NumberImpl</name>
<description>Implementation of the
com.ibm.uima.an_regex.String2Number interface</description>
<type>String</type>
<multiValued>false</multiValued>
<mandatory>true</mandatory>
 </configurationParameter>

 <configurationParameterSettings>
  <nameValuePair>
<name>String2NumberImpl</name>
   <value>
<string>com.ibm.uima.an_regex.impl.String2Number_impl</string>
   </value>
  </nameValuePair>
 </configurationParameterSettings>
</configurationParameters>
```

功能

正規表示式註解程式的輸入與輸出功能，以及其支援的語言定義在註解程式描述子的功能區段中。

描述子檔案中的輸入功能 (輸入類型) 必須符合規則集檔案中使用的相符類型。如果規則只使用 uima.tt.DocumentAnnotation 類型，則因爲此類型都是已定義的，所以您不需要宣告任何輸入功能。您必須定義所有其他類型。

由正規表示式註解程式所建立的註解類型指定在輸出功能區段中。這些類型必須符合在規則集檔案中宣告的輸出類型。

因爲正規表示式註解程式與語言無關，請指定代表任何語言的 x-unspecified。

類型系統說明

正規表示式註解程式 XML 描述子中的類型系統說明區段定義註解程式使用的類型系統。用於規則集 XML 檔的類型，以及在註解程式描述子的輸入及輸出功能區段中提到的類型，都必須符合在類型系統說明中定義的類型。

範例

描述子的類型系統說明區段會匯入類型系統描述子 XML 檔：

```
<typeSystemDescription>
 <imports>
<import location="./xml/of_sample_regex_typesystem.xml"/>
 </imports>
</typeSystemDescription>
```

外部資源

描述子的外部資源區段包含註解程式所需的檔案及類別。

正規表示式註解程式需要規則集檔案。透過由類別 com.ibm.uima.an_regex.impl.FileResource_impl 實作的 com.ibm.uima.an_regex.FileResource 介面，可將規則集檔案用於正規表示式註解程式。若要將您的自訂規則傳遞至正規表示式註解程式，您必須在註解程式描述子中提供規則集檔案的名稱，並將檔案位置新增至您的類別路徑。正規表示式註解程式用於存取規則集檔案的金鑰名爲 RuleSetDefinition。請不要變更此金鑰，否則，正規表示式註解程式將找不到規則集，且註解程式將無法進行起始設定。

範例

描述子的外部資源區段如下所示：

```
<externalResourceDependencies>
 <externalResourceDependency>
<key>RuleSetDefinition</key>
<description>Rule set definition</description>
<interfaceName>com.ibm.uima.an_regex.FileResource</interfaceName>
<optional>false</optional>
 </externalResourceDependency>
</externalResourceDependencies>
<resourceManagerConfiguration>
 <externalResources>
  <externalResource>
<name>of_samples_regex_rules</name>
```

```
<description>Rule set definition file for room numbers</description>
   <fileResourceSpecifier>
<fileUrl>file:of_samples_regex_rules.xml</fileUrl>
   </fileResourceSpecifier>
   <implementationName>
    com.ibm.uima.an_regex.impl.FileResource_impl</implementationName>
  </externalResource>
 </externalResources>
 <externalResourceBindings>
  <externalResourceBinding>
<key>RuleSetDefinition</key>
<resourceName>of_samples_regex_rules</resourceName>
  </externalResourceBinding>
 </externalResourceBindings>
</resourceManagerConfiguration>
```

相關概念

第 73 頁的『正規表示式註解程式』
正規表示式註解程式可讓您執行自訂文字分析，而無需實作您自己的文字分析引擎。根據您可以自行定義的一組規則 (正規表示式)，正規表示式註解程式會偵測純文字文件中的資訊結構，並在共用分析結構中建立已偵測之資訊的註解。

第 76 頁的『規則集檔案』
在正規表示式註解程式中，XML 規則集檔案會以正規表示式形式來定義用於剖析純文字文件的規則。

相關參考

『日誌記載』
所有正規表示式註解程式的日誌訊息都會寫入現行集合的日誌檔中。

日誌記載

所有正規表示式註解程式的日誌訊息都會寫入現行集合的日誌檔中。

集合日誌檔位於 ES_NODE_ROOT/logs/，且名稱格式爲 <*collection_id*>_<*current_date*>.log。您可以使用 esviewlogs.sh/.bat Script 來檢視日誌檔。

可能的記載層次有七種：

- 錯誤
- 警告
- 資訊
- 配置
- 細微
- 較細微
- 最細微

您無法變更「錯誤」及「警告」訊息的對映。根據預設值，只會將「資訊」、「警告」及「錯誤」訊息寫入日誌檔。這些是企業搜尋使用的標準記載層次。其他記載層次可以對映更多的詳細資訊。

若要接收正規表示式註解程式的日誌訊息，則必須至少將記載層次設爲「配置」。在此層次中，註解程式會記載配置設定，例如使用的規則集檔案及 com.ibm.uima.an_regex.String2Number 介面的實作類別名稱。

例如，如果將記載層次設爲「較細微」，則註解程式會記載無法建立的註解。這可協助您判定無法建立所有您希望建立之註解的原因。例如，可能是其中一個正規表示式中含有錯誤，或是選用的擷取群組與文件中的某個文字不相符。類似地，如果指定將一個功能設爲符合擷取群組的文字順序，但沒有相符的文字順序，則會將特性設爲空值。

如需更詳細的資訊，請將記載層次設爲「最細微」。在此層次上，註解程式會記載現行正規表示式型樣、目前正在分析的部分文件文字，以及所有已建立的註解及特性。使用更詳細的記載，特別是記載層次「較細微」及「最細微」，會降低註解程式的整體效能。

例如，如果您需要詳細的記載層次對映，請將配置設定 `trevi.tokenizer.jedii.InformationalLevelMapping=Info` 變更爲 `trevi.tokenizer.jedii.InformationalLevelMapping=Finest`，來修改 *ES_NODE_ROOT*/master_config/parserservice/ 中稱爲 tokenizer.properties 的配置檔。

若要啓動記載層次變更，您必須使用管理主控台停止所有剖析器程序。然後，您必須從指令行停止剖析器服務階段作業，再重新啓動它，方法是呼叫：

```
>esadmin session parserservice stop
>esdamin session parserservice start
```

此後，可以再次啓用剖析，並且您現在應該已具有新的記載層次。每次變更記載層次時，您都必須重複這些步驟。

相關概念

第 73 頁的『正規表示式註解程式』
正規表示式註解程式可讓您執行自訂文字分析，而無需實作您自己的文字分析引擎。根據您可以自行定義的一組規則 (正規表示式)，正規表示式註解程式會偵測純文字文件中的資訊結構，並在共用分析結構中建立已偵測之資訊的註解。

第 76 頁的『規則集檔案』
在正規表示式註解程式中，XML 規則集檔案會以正規表示式形式來定義用於剖析純文字文件的規則。

相關參考

第 80 頁的『註解程式描述子』
正規表示式註解程式 XML 描述子包含執行註解程式所需之正規表示式註解程式的相關敘述性資訊。

企業搜尋文件

您可以閱讀 PDF 或 HTML 格式的 OmniFind Enterprise Edition 文件。

OmniFind Enterprise Edition 安裝程式會自動安裝 IBM Content Discovery 資訊中心，其中包括 HTML 版的 OmniFind Enterprise Edition 8.4 版及 WebSphere Information Integrator Content Edition 8.3 版產品文件。在多部伺服器安裝架構中，資訊中心會安裝在所有搜尋伺服器上。如果沒有安裝資訊中心，當您按一下說明時，資訊中心會開啓在 IBM 網站上。

若要參閱已安裝的 PDF 版文件，請移至 ES_INSTALL_ROOT/docs/*locale*/pdf。例如，若要尋找英文版文件，請移至 ES_INSTALL_ROOT/docs/en_US/pdf。

若要存取所有可用語言的 PDF 版文件，請參閱 OmniFind Enterprise Edition 8.4 版文件站台。

您也可以從 OmniFind Enterprise Edition Support 站台，存取產品下載、修正套件、TechNotes 以及資訊中心。

下表顯示可用的文件、檔名及位置。

表 12. 企業搜尋的說明文件

標題	檔名	位置
資訊中心		http://publib.boulder.ibm.com/infocenter/discover/v8r4/
企業搜尋安裝手冊	iiysi.pdf	ES_INSTALL_ROOT/docs/*locale*/pdf/
快速入門手冊 (此文件也有英文、法文及日文的印刷版本)。	QuickStartGuide_*two-letter locale*.pdf	ES_INSTALL_ROOT/docs/*locale*/pdf/
企業搜尋安裝基本需求	iiysr.txt 或 iiysr.htm	ES_INSTALL_ROOT/docs/*locale*/ (您也可以從安裝啓動程式存取此檔案)
管理企業搜尋	iiysa.pdf	ES_INSTALL_ROOT/docs/*locale*/pdf/
Programming Guide and API Reference for Enterprise Search	iiysp.pdf	ES_INSTALL_ROOT/docs/en_US/pdf/
疑難排解指南及訊息參考手冊	iiysm.pdf	ES_INSTALL_ROOT/docs/*locale*/pdf/
文字分析整合	iiyst.pdf	ES_INSTALL_ROOT/docs/*locale*/pdf/
Plug-in for Google Desktop Search	iiysg.pdf	ES_INSTALL_ROOT/docs/*locale*/pdf/
版本注意事項	iiysn.pdf	可在 OmniFind Enterprise Edition 8.4 版文件站台取得 (您也可以從安裝啓動程式存取此檔案)

WebSphere Information Integrator OmniFind Edition 協助工具

可以存取 IBM WebSphere Information Integrator OmniFind Edition 使用者介面及文件。

安裝程式

您可以使用快速鍵，在 WebSphere Information Integrator OmniFind Edition 安裝程式中瀏覽及執行動作。下表說明部分快速鍵。

表 13. 安裝程式的鍵盤快速鍵

動作	快速鍵
標示圓鈕	方向鍵
選取圓鈕	Tab 鍵
標示按鈕	Tab 鍵
選取按鈕	Enter 鍵
前往下一個或前一個視窗，或者取消	按下 Tab 鍵以標示出某一按鈕，然後按 Enter 鍵
使作用中視窗停用	Ctrl + Alt + Esc

企業搜尋管理主控台及資訊中心

管理主控台及資訊中心介面是瀏覽器型介面，您可以使用 Microsoft® Internet Explorer 或 Mozilla FireFox 來檢視。如需瀏覽器的快速鍵及其他協助工具特性清單，請參閱 Internet Explorer 或 FireFox 的線上說明。

PDF 文件

您可以檢視所有 PDF 格式的企業搜尋文件。只要使用 Adobe Acrobat 6.0 版，即可存取 PDF 文件。PDF 文件已結構化，應該可以使用大部分的螢幕閱讀器來讀取。

企業搜尋術語的名詞解釋

此名詞解釋定義用於企業搜尋介面及文件的詞彙。

存取控制清單 (access control list, ACL)
此清單中包含一或多個使用者 ID 或使用者群組，以及它們的相關專用權。您可以使用存取控制清單來控制使用者對項目及物件的存取權。

管理角色 (administrative role)
決定使用者在企業搜尋管理主控台中可使用功能的使用者分類。此角色也決定使用者可以管理哪些集合。

分析引擎 (analysis engine)
請參閱文字分析引擎。

分析結果 (analysis results)
資訊由註解程式所產生。分析結果會寫入稱爲「共用分析結構」的資料結構中。自訂文字分析引擎 (註解程式) 所產生的分析結果，可以藉由併入企業搜尋索引，供進行搜尋。

註解 (annotation)
關於文字跨距的相關資訊。例如，註解可指出代表公司名稱的文字跨距。在「非結構化資訊管理架構 (UIMA)」中，註解是一種特殊的特性結構類型。

註解程式 (annotator)
執行特定語言分析作業，並產生和記錄註解的軟體元件。註解程式是分析引擎中的分析邏輯元件。

Boolean 搜尋 (boolean search)
使用運算子如 AND、NOT 及 OR 結合一或多個搜尋字詞的搜尋。

Boost 類別 (boost class)
可以影響文件在搜尋結果中相關性排序的一種規格。

Boost 字 (boost word)
可以影響文件在搜尋結果中相關性排序的單字。在查詢處理程序期間，含有 Boost 字的文件重要性會提高或降低，視該單字預先定義的評分而定。

種類 (category)
具相同內容的文件群組。

種類樹狀結構 (category tree)
顯示在企業搜尋管理主控台的種類階層結構。

認證 (certificate)
一種數位文件，可以將公開金鑰連結至憑證擁有人的身分，藉以鑑別憑證擁有人。憑證是由憑證管理中心發出。

憑證管理中心 (certificate authority)
發出憑證及鑑別電子交易中相關實體 (個人或組織) 的一種組織。憑證管理中心可保證交換資訊的雙方身分沒有問題。

字元正常化 (character normalization)
將字元的變體形式 (如大寫字體及區別發音符號) 簡化成一般形式的程序。

附著語素 (clitic)
語法上的功能完全不同，但在語音上連至另一個字的一種單字。附著語素的寫法可以和連結的單字相連接，或完全分開。附著語素的常見範例包括英文縮語的最後一部分 (*would**n't*** 或 *you**'re***)。

集合 (collection)
一組資料來源及搜索、剖析、建立索引及搜尋這些資料來源的選項。

共用分析結構 (common analysis structure, CAS)
此結構儲存文件的內容、中繼資料，以及由文字分析引擎產生的所有分析結果。文件分析期間的所有資料交換，都是使用「共用分析結構」來處理。

共用分析結構消費者 (CAS 消費者)
共用分析結構消費者會執行共用分析結構中儲存的分析結果之最終處理。例如，消費者會索引搜尋引擎中的共用分析結構內容，或將特定的分析結果移入關聯式資料庫。

共用通訊層 (common communication layer, CCL)
聯合 WebSphere Information Integrator OmniFind Edition 中各種元件 (控制器、剖析器、搜索器、索引伺服器) 的通訊架構。

概念萃取 (concept extraction)
定義文字文件的重要字彙項目 (例如人名、地點及產品)，以及產生這些項目的清單的文字分析函數。另請參閱主題萃取。

搜索範圍 (crawl space)
與搜索器爲建立索引時，從擷取項目讀取的指定型樣 (例如統一資源定位器 URL、資料庫名稱、檔案系統路徑、網域名稱及 IP 位址) 相符的一組來源。

搜索器 (crawler)
從資料來源擷取文件，以及收集可用於建立搜尋索引的資訊的軟體程式。

認證 (credential)
鑑別時獲得的詳細資訊，說明使用者、所有群組關聯及其他與安全相關的身分屬性。認證可用來執行許多服務，如授權、審核及委任。

自訂文字分析引擎 (custom text analysis engine)
使用「非結構化資訊管理架構 (UIMA)」軟體開發套件 (SDK) 所建立的文字分析引擎，並且可以新增至一組標準企業搜尋文字分析引擎中 (也稱爲企業搜尋基礎註解程式)。另請參閱文字分析引擎。

資料來源 (data source)
可擷取文件的任何資料儲存庫，例如 Web、關聯式及非關聯式資料庫、以及內容管理系統。

資料來源類型 (data source type)
依據存取資料通訊協定的資料來源分組。

資料儲存庫 (data store)
其中的文件資料保留其剖析格式的資料結構。剖析器會寫入資料儲存庫。資料儲存庫是用來建立索引，以及產生搜尋摘要。它和原始資料儲存庫不同。

差異索引建置 (delta index build)
將新資訊新增至企業搜尋系統現存索引的程序。請對照主索引建置。

移出佇列 (dequeue)
從佇列中移除項目。

區別發音符號 (diacritics)
加入字母以變更單字發音或區別類似單字的一種標記，如重音符號或德文曲音。

探索器 (discoverer)
搜索器的一項功能，決定哪些資料來源可讓搜索器擷取資訊。

識別名稱 (distinguished name)
唯一識別目錄項目的名稱。識別名稱是由逗點區隔的屬性:值組組成。另外，一組名稱值組 (如 CN=人名，及 C=國家或地區) 可唯一識別數位憑證的實體。

文件物件模型 (Document Object Model)
以物件樹方式檢視結構化文件 (如 XML 檔) 的系統，可以利用程式化的方式來存取及更新文件。

Domino® Document Manager 檔案櫃 (Domino Document Manager cabinet)
用來組織文件的 Domino Document Manager 資料庫。檔案櫃保留 Domino 資料庫。

Domino Document Manager 檔案庫 (Domino Document Manager library)
Domino Document Manager 資料庫，它是 Domino Document Manager 的進入點。

Domino Internet Inter-ORB Protocol (DIIOP)
在伺服器上執行的一種伺服器作業，可以搭配 Domino Object Request Broker 一起使用，以便在使用 Notes® Java 類別所建立的 Java Applet 與 Domino 伺服器之間通訊。瀏覽器使用者及 Domino 伺服器均使用 DIIOP 來通訊及交換物件資料。

動態排序 (dynamic ranking)
查詢中詞彙的排序類型，這些詞彙是用來分析要搜尋的相關文件，以決定結果的排序。另請參閱文字計分。反義詞爲靜態排序。

動態彙總 (dynamic summarization)
搜尋字詞強調顯示，且搜尋結果包含最能代表使用者搜尋文件概念的詞組的彙總類型。反義詞爲靜態彙總。

加入佇列 (enqueue)
在佇列中加入項目。

企業搜尋管理者 (enterprise search administrator)
讓使用者可以管理整個企業搜尋系統的管理角色。

企業搜尋基礎註解程式 (enterprise search base annotators)
一組標準文字分析引擎，在企業搜尋中，用來進行預設文件分析處理。

跳出字元 (escape character)
爲之後的一或多個字元抑制或選取特殊意義的字元。

外部資料來源 (external data source)
用於聯合，且未由 WebSphere Information Integrator OmniFind Edition 搜索、剖析或索引的資料來源。外部資料來源的搜尋委託給那些資料來源的查詢應用程式程式設計介面。

功能路徑 (feature path)
用來存取「非結構化資訊管理架構 (UIMA)」特性結構中特性值的路徑。

特性結構 (feature structure)
代表文字分析結果的基礎資料結構。特性結構是屬性值結構。每一個特性結構都屬於某一類型，且每一類型都已指定一組有效的特性或屬性，非常類似 Java 類別。

聯合搜尋 (federated search)
可以在多個搜尋服務上執行搜尋，並傳回搜尋結果合併清單的一種搜尋功能。

聯合功能 (federation)
合併命名系統的程序，使聚集系統可以處理跨命名系統的複合式名稱。

欄位 (field)
記錄中最小的可識別部分。

限定欄位搜尋 (fielded search)
限於某個特定欄位的查詢。

任意文字搜尋 (free text search)
搜尋時，搜尋字組以沒有格式的文字表示。

全文索引 (full text index)
參照資料項目以方便快速尋找內含查詢字詞的文件的資料結構。

模糊搜尋 (fuzzy search)
傳回與搜尋字詞拼法相似的字詞的搜尋。

混合式搜尋 (hybrid search)
結合 Boolean 搜尋及任意文字搜尋。

身分識別管理 (identity management)
使用原生存取控制來驗證使用者現行認證的能力。如果資料來源受到支援單一登入 (SSO) 鑑別的產品保護，並且已配置搜索器使用 SSO 安全機制，就會使用 SSO 機制來鑑別使用者。否則，會將使用者認證加密在安全的儲存區中，並在原生存取控制變更時，加以更新。

索引 (index)
請參閱全文索引。

索引佇列 (index queue)
待處理的主要及差異索引建置要求清單。

資訊萃取 (information extraction)
一種概念萃取的類型，可自動辨識文字文件中重要的詞彙項目，如名稱、術語及表示式。

IP 位址 (IP address)
唯一的 32 位元位址，用來識別網路上的主機。

Java 資料庫連線功能 (Java Database Connectivity, JDBC)
Java 平台及大量資料庫之間資料庫獨立連線功能的業界標準。 JDBC 介面提供 SQL 型資料庫存取的呼叫層次。

JavaScript™
在瀏覽器及 Web 伺服器中使用的一種 Web Scripting 語言。

JavaServer Pages (JSP)
一種伺服器 Scripting 技術，可以讓 Java 程式碼動態內嵌於網頁 (HTML 檔) 並在使用該網頁時執行，以便將動態內容傳回用戶端。

Java 虛擬機器 (Java virtual machine, JVM)
處理器的一種軟體實作，用來執行已編譯的 Java 程式碼 (Applet 及應用程式)。

Katakana
由兩種通用的日文語音字母之一所使用的符號組成的一種字集，主要用來依語音撰寫外文。

金鑰儲存庫檔案 (keystore file)
一種資料庫檔案，其中含有儲存作爲簽章者憑證的公開金鑰，以及儲存在個人憑證中的私密金鑰。

語言識別 (language identification)
決定文件語言的一種企業搜尋功能。

詞形 (lemma)
字的基礎詞形。在高度變音的語言中，詞形是很重要的，如捷克語。

詞形還原 (lemmatization)
查閱定義檔中指定單字詞形的程序。詞形還原不同於詞根索引 (詞根索引是一種演算法)，通常不使用列出語言單字的定義檔。

語彙聯繫關係 (lexical affinity)
在文件中彼此意義相近之搜尋單字的關係。語彙聯繫關係用於計算結果的關聯性。

檔案庫 (library)
一種系統物件，當成其他物件的目錄使用。另請參閱 Domino Document Manager 檔案庫。

連字 (ligature)
兩個以上的連接字元，使它們看起來像是同一個字，如結合 a 和 e 形成連字 æ。

輕量型目錄存取通訊協定 (Lightweight Directory Access Protocol, LDAP)
一種開放式通訊協定，使用 TCP/IP 存取支援 X.500 模型的目錄，且不需要更複雜的 X.500 Directory Access Protocol 的資源需求。

語言學搜尋 (linguistic search)
一種搜尋類型，利用還原成基礎詞形 (例如，*mice* 以 *mouse* 索引) 或從基礎詞形擴充 (如複合字) 的詞彙來瀏覽、擷取及索引文件。

鏈結分析 (link analysis)
以文件間超鏈結分析爲基礎的一種方法，用來決定集合中的哪些頁面對使用者是重要的。

本端聯合器 (local federator)
聯合一組可搜尋物件的用戶端聯合器。

Lotus® QuickPlace® 工作區 (Lotus QuickPlace place)
由 Lotus QuickPlace 提供的一種 Web 集合點，可以讓散佈在不同地理環境的參與者在專案中分工合作，並在結構化的安全工作區中連線通訊。

Lotus QuickPlace 檔案室 (Lotus QuickPlace room)
Lotus QuickPlace 工作區的分割區，限於需要集體分擔工作的授權成員使用。

主索引建置
建置整個企業搜尋系統的完整索引的程序。請對照差異索引建置。

遮罩字元 (masking character)
用於代表搜尋字詞前面、中間及尾端選用字元的字元。遮罩字元通常用於尋找索引中詞彙的變體。另請參閱萬用字元。

MIME 類型 (MIME type)
一種網際網路標準，用來識別要在網際網路上傳送的物件類型。

監督者 (monitor)
一種企業搜尋使用者，此使用者具有觀察集合層次程序的權限。

自然語言查詢 (natural language query)
分析書面詞句 (例如 ″Who runs the finance department?″) 而非簡單關鍵字集合的搜尋類型。

換行字元 (newline character)
造成列印或顯示位置向下移一行的一種控制字元。部分系統需要多個字元。

n-gram 斷詞法 (n-gram segmentation)
一種分析方法，會將指定字元數的重疊順序視爲一個單字，而不是像 Unicode 型空格斷詞法使用空格來區隔單字。

no-follow 指引 (no-follow directive)
網頁中的一種指引，指示 robot (如 Web 搜索器) 不遵循在那些網頁中找到的鏈結。

no-index 指引 (no-index directive)
網頁中的一種指引，指示 robot (如 Web 搜索器) 不在索引中併入那些網頁的內容。

Notes 遠端程序呼叫 (Notes remote procedure call, NRPC)
Lotus Notes® 的通訊機制，用於所有 Notes 對 Notes 的通訊。

操作員 (operator)
一種企業搜尋使用者，具有觀察、啓動及停止集合層次程序的權限。

參數搜尋 (parametric search)
一種搜尋類型，用來在指定範圍內尋找含有數值或屬性 (如日期、整數或其他數值資料類型) 的物件。

剖析器 (parser)
解譯新增至企業搜尋資料儲存區的文件的程式。剖析器會從文件中取出資訊，並準備好以供建立索引、搜尋及擷取之用。

剖析器驅動程式 (parser driver)
供給剖析器服務所需文件的企業搜尋服務。每一個集合會有一個剖析器驅動程式。集合的剖析器驅動程式服務，會和該集合在企業搜尋管理主控台中的剖析器相對應。

剖析器服務 (parser service)
一種企業搜尋服務，會處理所有文件剖析及所有文件集合的文字分析處理。至少會有一項剖析器服務在執行中。

工作區 (place)
可以在入口網站中看見的虛擬位置，個人和群組可以在該處協商分工合作。在入口網站中，每一個使用者都有私人工作用的個人工作區，而群組可以存取各種共用工作區 (可以是公用位置或限制工作區)。另請參閱 Lotus QuickPlace 工作區。

熱門等級 (popular ranking)

依據文件的熱門程度新增至文件現有等級的排序類型。

Portal Document Manager (PDM)

容許使用者擁有一座團隊協同作業使用的核心文件儲存庫。讓管理者能夠有效地管理他們的文件，並能控制使用者與資訊互動的方式。

處理程序引擎保存檔 (processing engine archive)

一種 .pear zip 保存檔，其中含有「非結構化資訊管理架構 (UIMA)」分析引擎以及在企業搜尋中用來進行自訂分析所需的所有資源。

鄰近搜尋 (proximity search)

在相同句子、段落或文件中尋找特定單字的搜尋類型。

Proxy 伺服器 (proxy server)

一種伺服器，當成應用程式或 Web 伺服器所管理的 HTTP Web 要求媒介使用。 Proxy 伺服器可以當成企業中內容伺服器的代用品。

快速鏈結 (quick link)

統一資源識別碼 (URI) 及關鍵字或詞組之間的關聯。

排序 (ranking)

爲查詢的搜尋結果中每一份文件指定一個整數值的程序。搜尋結果中的文件次序是依據查詢的相關性。較高的等級表示較爲相符。另請參閱動態排序及靜態排序。

原始資料儲存庫

在搜索到的文件傳送到剖析器之前，所存放的資料結構。搜索器會寫入原始資料儲存庫，而剖析器會讀取原始資料儲存庫。當文件剖析完成後，就會從原始資料儲存庫中移除文件。它和資料儲存庫不同。

正規表示式註解程式 (regular expression annotator)

正規表示式註解程式會偵測在純文字文件中的實體或單位，例如電話號碼、產品編號、員工姓名或地址，它會根據說明在文件文字中進行搜尋的確切型樣之正規表示式作偵測。如果其中一項正規表示式符合部分文件文字，則正規表示式註解程式會建立相對應的註解程式，來涵蓋相符項或它的一部分。接著，它會使用索引對映檔，將這些註解的表示式儲存在企業搜尋中，或是使用資料庫對映檔，儲存在可處理 JDBC 的資料庫中。

遠端聯合器 (remote federator)

聯合一組可搜尋物件的伺服器聯合器。

Robots Exclusion Protocol

一種通訊協定，可以讓網站管理者指示站台的哪些部分不得讓 robot 造訪。

檔案室 (room)

一種程式，可讓使用者建立文件以供他人讀取、回應他人意見，以及檢視專案狀態和截止時間。使用者也可以和其他在同一檔案室中的使用者聊天。另請參閱 Lotus QuickPlace 檔案室。

規則種類 (rule-based category)

依照規則建立的種類，指定哪些文件與哪些種類相關聯。例如，您可以定義規則，設定內含或不含特定單字或符合統一資源識別碼 (URI) 型樣的文件，與特定的種類相關。

範圍 (scope)
一組相關的統一資源識別碼 URI，用來定義搜尋要求的範圍。

搜尋應用程式 (search application)
在企業搜尋系統中，爲集合處理查詢、搜尋索引、傳回搜尋結果、以及擷取來源文件的程式。

搜尋快取 (search cache)
保留資料及先前搜尋要求的結果的緩衝區。

搜尋引擎 (search engine)
接受搜尋要求並將文件清單傳回給使用者的程式。

搜尋索引檔 (search index files)
一組檔案，其中的索引儲存於搜尋引擎。

搜尋結果 (search results)
符合搜尋要求的文件清單。

Secure Sockets Layer (SSL)
提供通訊私密性的一種安全通訊協定。

安全記號 (security token)
關於身分及安全的資訊，用於對集合中文件的存取授權。不同的資料來源類型支援不同的安全記號類型。範例包括使用者角色、使用者 ID、群組 ID、以及可用於控制內容存取的其它資訊。

種子清單頁面 (seed list page)
在 WebSphere Portal 中的一個 XML 頁面，其中包含與入口網站中的頁面鏈結。搜索器會使用該種子清單來識別要搜索的文件。種子清單頁面中，也會包含與企業搜尋索引中已搜索文件一起儲存的中繼資料。

啓動統一資源定位器 (URL)
搜索的開始點。

斷詞法 (segmentation)
將文字分割成明確的詞元。非字典式的處理包括空格及 n-gram 斷詞法，而字典式支援包括單字、句子、段落斷詞法及詞形還原。

語意搜尋 (semantic search)
語意搜尋藉由納入更多與語言及搜尋解決方案網域相關的知識，來加強關鍵字搜尋參照範例。包含及應用此知識的技術稱爲文字分析。

servlet
在 Web 伺服器上執行的一種 Java 程式，可以產生回應 Web 用戶端要求的動態內容以延伸伺服器功能。 Servlet 通常是用來連接資料庫與 Web。

排列組合 (shingle)
從一段句子取出的一串連續記號 (單字)。例如，從 ″This is a very short sentence.″ 中，三個字的排列組合 (或稱爲三字) 爲：

This is a
is a very
a very short
very short sentence

排列組合可以用在統計語言學中。例如，如果兩個不同文字有很多共同的排列組合，這些文字可能某種程度上是有相關的。

軟錯誤頁面 (soft error page)
一種特殊頁面，可在 HTTP 伺服器無法傳回用戶端所要求的頁面時詳細說明問題，並配置 HTTP 伺服器傳回這些頁面，而不是只在回應中顯示標頭及指出發生什麼問題的回覆碼。

靜態排序 (static ranking)
一種排序類型，以有關要排序文件的因數 (如日期、指向文件的鏈結數等等) 來提高排序。反義詞爲動態排序。

靜態彙總 (static summarization)
搜尋結果包含指定的已儲存文件摘要的彙總類型。反義詞爲動態彙總。

詞幹 (stemming)
請參閱字根索引。

停止單字 (stop word)
搜尋應用程式不處理的一種常用單字，如 *the*、*an* 或 *and*。

停止單字移除 (stop word removal)
從查詢中移除停用字的程序，以忽略常用字並傳回更多相關結果。

彙總 (summarization)
在搜尋結果中包括句子的程序，以簡短地描述文件的內容。請參閱動態彙總及靜態彙總。

同義字定義檔 (synonym dictionary)
一種定義檔，可以讓使用者在搜尋集合時搜尋查詢字詞的同義字。

分類法 (taxonomy)
依據相似性分出物件群組的分類。在企業搜尋中，分類法可組織資料入種類及子種類。另請參閱種類樹狀結構。

文字分析 (text analysis)
從文字中擷取語意及其他資訊以加強集合中資料可擷取性的程序。

文字分析引擎 (text analysis engine)
一種軟體元件，負責尋找及表示文字中的環境定義及語意內容。

文字計分 (text-based scoring)
爲文件指定整數值的程序，以表示文件與查詢字詞的相關性。較高的整數值表示與查詢較相符。另請參閱動態排序。

文字斷詞法 (text segmentation)
請參閱斷詞法。

主題萃取 (theme extraction)
概念萃取的一種類型，自動辨識文字文件中重要的字彙項目，以取出文件的主題。另請參閱概念萃取。

記號 (token)
企業搜尋索引的基本文字單元。記號可以是語言的單字或其他適用於索引的文字單元。

記號化 (tokenization)
請參閱斷詞法。

記號器 (tokenizer)
一種文字斷詞法程式，可掃描文字並判斷文字系列是否可以識別爲記號及其時機。

尾端字元 (trailing character)
位於單字中最後一個位置的字元。

類型系統 (type system)
類型系統定義可能會被文件中之文字分析引擎發現的物件 (特性結構) 類型。類型系統以類型和特性來定義所有可能的特性結構。您可以在類型系統中，定義任意數目的不同類型。類型系統和網域及應用程式攸關。

Unicode 型空格斷詞法 (Unicode-based white space segmentation)
記號化的方法，使用 Unicode 字元內容來區分記號及分隔字元之間。

統一資源識別碼 (Uniform Resource Identifier, URI)
識別抽象或實體資源的精簡字串。

統一資源定位器 (Uniform Resource Locator, URL)
在電腦或網路如網際網路上，代表資訊資源的一連串字元。此一連串字元包括用於存取資訊資源的通訊協定縮寫名稱，此由通訊協定使用以尋找資訊資源。

通用資源名稱 (Universal Resource Name, URN)
一種網際網路通訊協定元素，由符合特定語法的短字串組成。字串包含可用來參照資源的名稱。

非結構化資訊管理架構 (Unstructured Information Management Architecture, UIMA) 一種 IBM 架構，定義實作系統以進行非結構化資料分析的架構。

使用者代理程式 (user agent)
瀏覽 Web 並在其所造訪網站中留下自己的資訊的應用程式。在企業搜尋中，Web 搜索器是使用者代理程式。

Web 搜索器 (Web crawler)
Robot 軟體類別，可以擷取 Web 文件並遵循該文件中的鏈結以探索 Web。

加權字詞搜尋 (weighted term search)
某些字詞重要性較高的查詢。

萬用字元 (wildcard character)
用於代表搜尋字詞前面、中間或尾端選用字元的字元。

字根索引 (word stemming)
語言正常化的程序，將單字的變化形簡化成常見形。例如，*connections*、*connective* 及 *connected* 之類的單字會還原成 *connect*。

XML 路徑語言 (XML Path Language, XPath)
唯一識別或定址來源 XML 文件組件的語言。 XPath 也提供操作字串、數字和布林運算子的基本機能。

存取 Content Management and Discovery 的相關資訊

您可以透過電話或從 Web 上取得 IBM Content Management and Discovery 產品的相關資訊。

這裡所提供的電話號碼只適用於美國：

- 若要訂購產品或取得一般資訊：1-800-IBM-CALL (1-800-426-2255)
- 若要訂購書籍：1-800-879-2755

您可以從網址爲 http://www.ibm.com/software/sw-bycategory/subcategory/SWB40.html 的 Web 上找到 IBM Content Management and Discovery 產品的相關資訊。此站台包含可協助您進行下列作業的鏈結：

- 瞭解產品
- 購買產品
- 參與產品的試用版及測試版測試
- 取得產品支援

若要存取產品文件：

1. 請造訪 Web，網址爲 http://www.ibm.com/software/sw-bycategory/subcategory/SWB40.html。
2. 選取您要進一步瞭解的產品，例如，WebSphere Information Integrator OmniFind Edition。 此站台包含下列內容的鏈結：
   - 產品文件，包括版本注意事項及連線資訊中心
   - 系統需求
   - 產品下載
   - 修正套件
   - 產品新訊
   - 產品支援資料，如白皮書及 IBM Redbooks™
   - 新聞群組及使用者群組
   - 訂購書籍的指示
3. 按一下頁面左邊的 Support 鏈結。
4. 在 Learn 區段中，選取您要檢視的文件類型。如果有適用於所選產品的資訊中心，您可以選取該資訊中心的鏈結。

提供文件的相關意見

請將您對本資訊或其他 IBM 文件的任何意見傳送給我們。

您的意見有助於 IBM 提供高品質的資訊。如果您對本資訊或其他隨附於 IBM Software Development 產品的文件有任何意見，請將您的意見傳送給我們。您可以使用下列任一方法來提供意見：

1. 您可以利用 www.ibm.com/software/awdtools/rcf/ 的線上讀者意見表，來傳送您的意見。

2. 請將您的意見以電子郵件寄至 comments@us.ibm.com。請併入產品名稱、產品的版本號碼，以及書籍資訊的名稱與產品編號 (如果有的話)。如果您對特定文字有意見，請說明該文字的位置 (例如，標題、表格號碼或頁碼)。

聯絡 IBM

若要聯絡美國或加拿大的 IBM 客戶服務中心，請撥打 1-800-IBM-SERV (1-800-426-7378)。

若要瞭解適用的服務選項，請撥打下列其中一個號碼：

- 美國：1-888-426-4343
- 加拿大：1-800-465-9600

若要尋找您所在國家或地區的 IBM 辦事處，請參閱 IBM Directory of Worldwide Contacts 網站，網址爲 www.ibm.com/planetwide。

注意事項與商標

注意事項

本資訊是針對 IBM 在美國所提供之產品與服務開發出來的。IBM 不見得會對所有國家或地區都提供本文件所提的各項產品、服務或功能。要知道在您所在地區是否可得到這些產品及服務時，請向當地的 IBM 服務代表查詢。而此處任何對於 IBM 產品、程式或服務的參考之處，並不表示或暗示只可以使用 IBM 的產品、程式或服務。任何未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，使用者必須自行負責評估和驗證任何非 IBM 產品、程式或服務的作業。

在本文件中可能包含著 IBM 所擁有之專利或擱置專利申請的內容。本文件使用者並不享有前述專利之任何授權。您可以用書面方式來查詢授權，來函請寄到：IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

若要查詢二位元組 (DBCS) 資訊的授權事宜，請連絡您國家或地區的 IBM 智慧財產部門，或者用書面方式寄到：IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106-0032, Japan

下列段落不適用於英國或任何其他與當地法律相抵觸的國家或地區： IBM 公司係以『現狀』提供本出版品，且不作任何明示或默示的保證，包括但不僅限於非侵害、可售性或符合特定用途之暗示保證。有些地區不允許放棄在特定交易中的明示或默示保證，因此，這項聲明對您可能不適用。

本書中可能會有技術上的錯誤或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 得隨時修改及/或變更本書中所說明的產品及/或程式，恕不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，爲便利 貴客戶之使用，而非爲該網站背書。這些網站中的資料，並不包含在 IBM 產品的資料中，使用網站中的資料，須自行負擔風險。

在不造成您困擾或損及您個人權益的前提下，IBM 得以適切使用或散佈您以各種型式所提供的相關資訊。

本程式之獲授權者若希望取得本程式之相關資訊，以便達到下列目的：(i) 在獨立建立的程式與其他程式 (包括本程式) 之間交換資訊；以及 (ii) 相互使用已交換的資訊。則請與位於下列地址之人員連絡：

IBM Corporation J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

上述資料之取得有其條件，在某些情況下必須付費方得使用。

IBM 基於「IBM 客戶合約」、「IBM 國際程式授權合約」或雙方之間任何同等的合約等條款，提供本文件中所說的授權程式與其所有適用的授權資料。

任何此處涵蓋的執行效能資料都是在一個受控制的環境下決定出來的。因此，若在其他作業環境下，所得的結果可能會大大不同。有些測定已在開發階段系統上做過， 不過這並不保證在一般系統上會出現相同結果。再者，有些測定可能已透過推測方式評估過。但實際結果可能並非如此。 本文件的使用者應依自己的特定環境，查證適用的資料。

非 IBM 產品的相關資訊，取自該產品供應商、發佈的聲明或其他公共來源。IBM 未測試這些產品，因此無法確認非 IBM 產品的效能、相容性或其他聲明。有關非 IBM 產品的功能問題，請洽該產品供應商。

有關 IBM 未來動向的任何陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

這個資訊中包含每日業務使用的報告和資料範例。爲使說明盡可能完備，範例中包含個人、公司、品牌及產品的名稱。此等名稱皆屬虛構，凡有類似實際個人或企業所用之名稱及地址者，皆屬巧合。

著作權授權：

本資訊可包含原始語言的範例應用程式，用以說明各種作業平台上的程式設計技術。 貴客戶得爲開發、使用、行銷或散佈運用樣本程式之作業平台的應用程式程式介面所撰寫的應用程式之目的，免費複製、修改並散佈這些樣本程式。此些範例並未在所有情況下完整測試。故 IBM 不保證或默示保證這些樣本程式之可靠性、服務性或功能。 貴客戶得爲開發、使用、行銷或散佈符合 IBM 應用程式設計介面的應用程式之目的，免費複製、修改並散佈這些樣本程式。

這些範例程式的每個複本或任何部分，或任何衍生作品都必須包括以下版權聲明：

Outside In (®) Viewer Technology, © 1992-2006 Stellent, Chicago, IL., Inc. All Rights Reserved.

IBM XSLT Processor Licensed Materials - Property of IBM ©Copyright IBM Corp., 1999-2006. All Rights Reserved.

商標

本主題列出 IBM 商標及某些非 IBM 商標。

如需 IBM 商標的相關資訊，請參閱 http://www.ibm.com/legal/copytrade.shtml。

下列術語是其他公司的商標或註冊商標：

Java 及所有以 Java 爲基礎的商標和標誌是 Sun Microsystems, Inc. 在美國及 (或) 其他國家的商標或註冊商標。

Microsoft、Windows、Windows NT 以及 Windows 標誌是 Microsoft Corporation 在美國及 (或) 其他國家的商標。

Intel、Intel Inside (標誌)、MMX 及 Pentium 是 Intel Corporation 在美國及 (或) 其他國家的商標。

UNIX 是 The Open Group 在美國及其他國家的註冊商標。

Linux 是 Linus Torvalds 在美國及 (或) 其他國家的商標。

其他公司、產品或服務名稱，可能是其他公司的商標或服務標誌。

索引

索引順序以中文字，英文字，及特殊符號之次序排列。

〔四劃〕

〔五劃〕

〔六劃〕

〔八劃〕

〔十一劃〕

〔十二劃〕

〔十三劃〕

〔十四劃〕

〔十五劃〕

〔十八劃〕

B

D

E

N

O

S

U

讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；懇請您閱讀後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號 (∨)；我們會在下一版中，作適當修訂，謝謝您的合作!

評估項目	評 估 意 見		備 註
正 確 性	內容說明與實際程序是否符合	□是□否	
	參考書目是否正確	□是□否	
一 致 性	文句用語及風格，前後是否一致	□是□否	
	實際產品介面訊息與本書中所提是否一致	□是□否	
完 整 性	是否遺漏您想知道的項目	□是□否	
	字句、章節是否有遺漏	□是□否	
術語使用	術語之使用是否恰當	□是□否	
	術語之使用，前後是否一致	□是□否	
可 讀 性	文句用語是否通順	□是□否	
	有否不知所云之處	□是□否	
內容說明	內容說明是否詳盡	□是□否	
	例題說明是否詳盡	□是□否	
排版方式	本書的形狀大小，版面安排是否方便閱讀	□是□否	
	字體大小，顏色編排，是否有助於閱讀	□是□否	
目錄索引	目錄內容之編排，是否便於查找	□是□否	
	索引語錄之排定，是否便於查找	□是□否	
	※評估意見為 "否" 者，請於備註欄提供建議。		

其他： (篇幅不夠時，請另外附紙說明。)

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。

註： 您也可將寶貴的意見以電子郵件寄至 tscadmin@tw.ibm.com，謝謝。

IBM OmniFind Enterprise Edition
8.4 版

SC40-2071-01

文字分析整合

折疊線

110
台北市信義區松仁路 7 號 3 樓

大臺灣
中華國際商業機器股份有限公司
研發中心
軟體國際部
啟

寄

折疊線

廣告回信
台灣北區郵政管理局登記
北台字第 00176 號
（免貼郵票）

寄件人
姓名：
地址：

SC40-2071-01