歐洲專利局光碟資料庫研習報告

智慧財產局　邱淑玟
司春玉

壹、 緣起

專利資訊 - 世界最大的技術資料寶庫

專利權是法律賦予專利權人在特定的領域及時間排除他人未經授權而使用、製造及販賣其物品或技術的權利，但相對地，專利權人必須揭露其技術內容，始能獲得此一保障。專利制度旨在調和專利權人與社會大眾之利益；在專利權人方面，其研發及製造的投資可以獲得補償並強化市場地位；產業界方面，可獲益於最新產業技術資訊之流通、避免重複的研究開發工作；而社會全體，則可獲享科技進步的成果。發明促進了科技之進步，並增進市場的發達與暢通，藉著保障專利權人在特定期間與地域內防止他人侵害其權利，以交換產業技術公開，達到專利制度刺激研究及開發，增進人類文明進步之目的。

由於專利制度在本質上係以公開技術內容，以取得排他權，因此專利資料有以下幾個特點：

最新的產業技術資訊：

發明人（含法人及個人）通常不願將它們的發明公開，但是為了獲取法律保護，它們會儘早提出專利申請，通常這些資訊會在專利案提出申請十八個月後被公開，技術內容亦經常為首次公開的，因此，專利資訊經常是最新產業技術資訊的來源。

內容詳細的技術文獻：

專利說明書必須將所發明的技術內容清楚而詳盡地揭露，使熟習該技術之人能充分瞭解，在目錄或論文中有時僅用數行描述一項產品，相對地在專利文獻可能有二十頁以上，這項嚴格的要求解釋了何以有百分之七十以上的實用技術資訊，除專利文獻外，他處無法獲得。。

具流通性：

所有的專利文獻均有一致的書目資料，且專利資料均依據國際公認的國際專利分類系統〈International Patent Classification〉分類，將各項極具價值或具戰略性的技術資訊分在大約七萬個類別中。檢索者可藉由分類或書目檢索到資料，再進而取得文獻。

容易取得的：

許多國家專利局檔案中蒐存完整的專利資料，參酌適當

的參考文獻，便可以在短時間內獲得所需資訊，比起訂購論文資料有時須數週之久，相對地方便許多。

涵蓋範疇廣：

全球每年被公開的專利案件約有一百萬件，其範圍包括了所有的科技領域，近年來甚至包括了生物科技、電腦軟體、商業方法等主題。

專利資料不僅有許多解決技術問題的資訊，它還包括了無窮盡的知識，據估計，有超過百分之八十的技術資訊可以在專利文獻中獲得。但是這一個技術及科學知識的寶庫多年來並未被充分利用。在歐洲據估計，過去五年裡，大約僅有五萬九千家公司曾使用專利系統，剩下約有十一萬多家公司應可利用專利系統，但未利用。其結果是已存在的發明，仍一再被投注人力、物力去從事發明；已經解決的問題再被努力解決、已有人製造的產品又被研究製造，這些重複的投資及研究，在歐洲經統計每年約浪費掉二百億美金。

因此所有專利主管機關均有責任，必須將專利資訊加以散播於公眾。但專利資料數量龐大，傳播上存在許多困難，例如紙本專利資料僅有圖書館蒐存，光碟資料庫亦僅有少數圖書館或專利資料中心提供公眾檢索，線上資料庫之檢索及資料列印費對於個人或

小公司而言過於昂貴，又專利資料的語文多樣，凡此種種皆導至專利資訊未能充份利用，所以必須研究一途徑使專利資訊充份擴散。

資訊傳播的方式隨著科技之進步，從傳統上以紙本形態為主，進步至線上資料庫、光碟資料庫、網際網路資料庫。也就是必須將資訊轉為電子檔案，透過電子媒體及通訊設備加以擴散。但是在檢索上仍有一極大的問題，就是語文的多樣性，在檢索及閱讀上造成困難，因此，建立一個單一語文且涵蓋各國專利資料之資料庫, 遂成為各國專利單位亟欲達到之目標, 亦為全球各界之期望。而英文，不容否認地，為全球使用最普遍之語文。然而專利資料浩瀚如海，任何單一專利主管機關或公司均無法建立此資料庫，唯有透過合作, 由各國製作其本國專利英文資料庫, 再加以整合，始能達成。德、日、韓、中國大陸等國在多年前即相繼發行英文專利光碟資料庫，並以此互相交換資料。

本局自八十一年起，即開始將部份本國專利資料譯為英文，建立資料庫，初時僅英譯發明專利（限本國人申請案），資料完整性不足，無法與各國進行資料交換。因此建立符合國際水準之英文專利資料庫，作為與國外資料交換之標的，促進專利資訊交流及我國專利資料國際化實屬刻不容緩。

本局為再充實英譯專利資料內容，自八十七年起，開始辦理發明及新型英文專利資料英譯之工作，迄今資料量已達相當程度， 進而資料庫之利用即成為課題，除將已建妥之資料庫放置於網際網路供公眾使用外，與其他國家之資訊交流及發行光碟資料庫，亦為努力之目標。

歐洲專利局當前在專利資料庫製作、提供上執牛耳之地位，其提供之專利資料庫-INPADOC、Esp@cenet 資料庫涵蓋範圍廣大，世界無出其右。在前案檢索、線上申請、無紙化等課題上亦有極大成就，足堪本局借鏡。民國八十九年間，本局與歐洲專利局洽談資料交換事宜，該局以其專利資料庫與本國英文專利資料庫作資料交換，並同意將其與美國專利商標局、日本特許廳三邊共同發展之光碟檢索引擎－MIMOSA 供本局應用於光碟資料庫製作。承本部國際合作處之支助，同年十一月間派員赴歐洲專利局就 MIMOSA 之技術加以研習。

歐洲專利局簡介

歐洲專利局係依據歐洲專利協定（European Patent Convention, EPC）而成立之國際性專利組織, 該協定於 1973 年 10 月 5 日簽訂於慕尼黑，生效於 1977 年，總局設於德國慕尼黑，分局於荷蘭海牙，附屬辦公室分設於德國柏林及奧

地利維也納，歐洲專利局之設立目的在使發明可較簡單、便宜、可靠地經由單一申請程序在諸簽約國中獲得保護，為歐洲國家合作的典範。歐洲專利組織包括其立法機關、 管理委員會、行政部門及歐洲專利局。職員約 4700 人，來自各會員國。其中慕尼黑有 2152 人、海牙 1938 人、柏林　185 人、維也納 86 人。

歐洲專利局並非 EU 之機構，完全自籌財源，具有相當程度的自主權。其所有運作費用及預算均來自於申請費用及部份的專利年費（一定比例，其他則歸各國專利局）

該局目前共有 20 個會員國（如下所列），其專利權保護並可延伸至部份的中歐及東歐國家。

奧地利	AT	Austria
希臘	GR	Hellenic Republic
比利時	BE	Belgium
愛爾蘭	IE	Ireland
瑞士	CH	Switzerland
義大利	IT	Italy
塞普魯斯	CY	Cyprus
列支敦斯登	LI	Liechtenstien
德國	DE	Germany
盧森堡	LU	Luxembourg

丹麥	DK	Denmark
摩納哥	MC	Monaco
西班牙	ES	Spain
荷蘭	NL	Netherlands
芬蘭	FI	Finland
葡萄牙	PT	Portugal
法國	FR	France
瑞典	SE	Sweden
土耳其	TR	Turkey
英國	GB	United Kingdom

申請歐洲專利，可以英文、德文或法文中任一種語文提出申請案, 經單一審查程序, 即可在任一會員國中獲得專利保護，並可擴大至部份東歐國家。若想獲得多個歐洲國家的專利保護，以此途徑在程序上較為便利，費用較低，因此歐洲專利申請案急速膨脹；在 1998 年歐洲專利局共收到 113400 件申請案、1999 年為 121750 件申請案。

當一歐洲專利被授與時，即受到申請人指定國之法律保護，所受到的保護與各國之國內專利一樣，有效期間為 20 年，在某些情況下，部份醫藥與植物的專利可延展其專利期限。

歐洲專利協定並與專利合作條約（Patent Cooperation Treaty， PCT）結合，後者提供單一而簡單的申請程序，在

超過 100 個國家有效，經由 PCT 申請亦可獲得歐洲專利。

歐洲專利系統的優點

對申請人而言:

- 對於尋求在多國獲得專利保護的專利申請案為一節約金錢與時間之途徑。
- 單一的專利授與程序，可以在歐洲專利簽約國中獲得保護
- 強而有力的專利保護：歐洲專利均經確實的審查程序，在一些國家中僅須註冊程序即可獲得專利。

對簽約國而言:

- 合理化: 無需重複前案調查及審查工作
- 專利文獻的合作
- 合諧化、更有效的專利法

對專利資訊利用人的好處

- 取得最新技術資訊
- 促進技術移轉
- 活化技術及市場
- 避免 R & D 的重複

由於關鍵技術的激烈競爭，專利資訊愈形重要，歐洲專利局透過與各國專利局密切合作，提供了 ESPACE 光碟資料庫系列，INPADOC 線上資料庫及 esp@cenet® 網際網路資料庫等服務，使存在於專利文獻的科技知識能為公眾取得。

歐洲專利局的國際合作

歐洲專利也可依據專利合作條約（Patent Cooperation Treaty，PCT）之程序，經由國際專利申請程序獲得。歐洲專利局為經世界智慧財產權組織認可之國際檢索及預審機構。其1999年處理之PCT申請案（製作檢索報告書及預審）件數與其他國際檢索及預審機構比較如下：

處理國際檢索案件比較表（1999年）

國際檢索機構	處理之申請案件數	百分比
歐洲專利局	44,713	60.4
美國	14640	19.8
日本	6827	9.2
瑞典	4380	5.9
澳洲	1378	1.9
奧地利	965	1.3
俄羅斯聯邦	526	0.7
西班牙	356	0.5
中國大陸	231	0.3
韓國	4	<0.1

處理國際預審案件比較表（1999年）

國際預審機構	請求數	百分比
歐洲專利局	30801	57.8
美國	14218	19.8
日本	6827	9.2
瑞典	4380	5.9
澳洲	1378	1.9
奧地利	965	1.3
俄羅斯聯邦	526	0.7
西班牙	356	0.5
中國大陸	231	0.3
韓國	4	<0.1

其處理之檢索及預審件數均超過所有PCT案件之百分之五十。由此可得知歐洲專利局於世界專利之地位。

歐洲專利局並以訓練、專家支援 、提供資料等方式，支援各國專利系統之改進或現代化。

歐洲專利之前案檢索及審查

當申請案開始進行前案檢索程序時，專利審查官將進行資料檢索，將相關前案資料檢索出來（不限於專利文獻），並作成檢索報告書，將與該申請案之可專利性有關之文獻列示出來。申請十八個月後，再將專利申請案公開，檢索報告書通常與說明書同時公開，但十八個月屆滿時若檢索報告書尚未完成，則分別公開。在1998年，為了滿足申請人希望於十八個月內完成檢索之需求，歐洲專利局開始採取加速處理之措施，自1995年起，這些措施被稱之為PACE計畫，在1997年，由在海牙及柏林的1000位檢索審查官完成93000件檢索，其中有4900件檢索是由其DG 2部門，14700件來自於比利時、法國、希臘、荷蘭、瑞士及土耳其的國際申請案。.

在海牙及柏林的檢索審查員（約1000人）及在慕尼黑的審查員，其學歷均為歐洲會員國的大學畢業以，並在專業技術領域具有專長。工作時使用歐洲專利局正式語文（英文、法文及德文）， DG 1的新審查員在被指派到其專長之技術領域前，須先參加內部的訓練課程， 接著接受二年由資深審查員指導的進階在職訓練，其內容係其專長技術領域及歐洲專利局DG1部門長期之資料檢索經驗。歐洲專利局極重視人員的在職訓練，其審查員必須接受訓練，以保持與其專長領域之技術發展同步。歐洲專利局亦安排審查員

進行參訪、參加技術研討會或上課，另外，審查員也必須與檢索工具一起進步，DG1也辦理經常性的團體訓練課程以凝聚1000位審查員之經驗。 同樣的，DG2的BEST 審查員也與DG1的審查員交換檢索經驗與訓練。

貳、 INPADOC 系統簡介

INPADOC 資料庫係由歐洲專利局製作，是世界最大的專利資料庫，涵蓋六十五個國家或專利組織之資料，其中有 22 個國家之資料包含法律狀態資料。

其內容穩定地每星期更新及增加，每星期約有 25000 件至 40000 件來自各地的專利資料加入其資料庫。該資料庫提供的法律狀態及專利家族資料，若想獲知一專利在世界各國的法律狀態（是否有效），本資料庫有最詳盡的資料，提供非常有用的資訊。可透過 DIALOG 、STN 等資料庫系統檢索該資料庫。本國之英文專利資料庫亦經由此次之資料交換計畫，加入此一資料庫中。

參、 MIMOSA

研習 MIMOSA 光碟軟體之背景狀況：

MIMOSA 光碟軟體的問世，最早是基於歐洲專利局(EPO)、美國專利商標局(USPTO)及日本特許廳(JPO)三方的合作計畫項目之一，其目的是為將各國數量成長迅速的專利資料，在短時間內利用可攜性的媒體出版，供各界人士查閱內容，以達到資訊交流的效果，除此之外，這個軟體應具有能夠檢索、擷取資料的功能，同時亦需提供一個易學易用的使用者界面，另就經濟及便利的因素考量，光碟片是目前最理想的資料儲存媒體，因此 MIMOSA 就是在這種需求的情況下所研發出來的產品。

MIMOSA 軟體是委由法國 Jouve S.A.公司負責製作的，軟體標的對象定位於處理智慧財產權資料，既然是要提供給各國不同環境的使用者，因此大家必須遵循一個共同的資料標準格式，以求其一致性，經決議採用國際智慧財產權組織(WIPO)所訂定的各項標準。資料型態分為文字及影像兩種，文字使用 SGML 格式；影像則使用嵌入式之 TIFF 檔案格式。由於各國的專利資料的欄位和內

容，會依專利制度的不同而異，為能適合各種產品內容需求，因此 MIMOSA 是屬於較為通用式的套裝軟體，當軟體安裝完成在硬體設備後，可透過各項的條件及參數的設定，達到符合出版產品的製作光碟環境，軟體中還提供了資料檢覈的功能，透過系統產生的報表檔案，使用者可以偵察出並修正錯誤的資料，MIMOSA 技術開發成功迄今，各國已陸續定期出版了多項專利資料相關的光碟產品。

美、日、歐三方目前 MIMOSA 合作計畫雖已告階段性終止，美國已另行開發其他系統之資料庫，但 EPO 仍持續提昇 MIMOSA 軟體的研發工作，目前 EPO 各會員國及非會員國，有超過 50 個國家使用 MIMOSA 發行專利資料光碟，且現行日本之英文專利(PAJ)光碟，亦仍是使用本項軟體。本(資料服務)組之首要工作項目之一，是和各國的智慧財產權組織相互交換資料，綜觀本國目前尚無相仿之英文專利光碟或其他更佳媒體之產品，在國際間資料交流時，較難以對等的方式進行，鑑於 MIMOSA 有 EPO 之運作及其龐大的使用群，本組遂於與 EPO 資料交換合作案外，擴展合作內容，引進 MIMOSA 軟體，評估日後出版本國英文專利光碟資料庫之可行性。經透過駐奧地利代表處經濟組協助與歐洲專利局維也納分局洽商，該局允諾免費提供是項軟體供本局使用，因此派

員赴維也納分局研習，附錄 MIMOSA 操作手冊為研訓時使用教材，此次研習為一入門課程，為本局首次接觸到該軟體技術內容，雖礙於研習時間僅有五天，對該軟體只能重點式項目說明，雖未能自行上機操作，深度了解各項軟硬體設備環境，但已有初步之認識，對日後本局自行建置資料庫有所助益。

MIMOSA 操作手冊主要為六大章節如下：

第１章　　簡介

第２章　　資料

第３章　　製作軟體

第４章　　資料準備

第５章　　建立資料索引

第６章　　線上工具

肆、電子申請

歐洲專利局業已開始試行線上申請作業，名為 epoline，其目標係將所有專利流程之文件由紙本改為電子檔案，其檔案傳遞經由網際網路，因此特別注重其安全性，以確保申請人可以機密地、確實地、完全地傳送申請案，epoline 之目標在達到線上申請、線上獲知申請案進度、線上註冊、線上傳送檢索報告書及線上付費。

對專利資訊的使用者，最重要的是線上獲知申請案進度及線上註冊，在 2001 年這二者將與 esp@cenet 結合。使用者將可透過網際網路獲得存於歐洲專利局之 PHOENIX 系統內申請案中可公開部份檔案，印出所需資料或下載 PDF 檔案。

在目前紙本作業環境中，要申請一份案卷資料通常需二個星期以上才能獲得，並且增加許多行政負擔，在線上作業中，申請者可立即獲得檔案資料，列印或下載所需部份，省下二個星期的等待時間及影印費用，歐洲專利局亦省下行政費用。

該局已完成 PHOENIX 回溯檔案掃描作業，此項服務將可提供歐洲專利局所有檔案。

對於申請人及大眾，EPOLINE 提供以下幾點好處：

1. 立即的回饋及確認
2. 快速提供檢索結果及審查結果
3. 可以在任何時間任何地點獲知狀態資料
4. 節省紙上作業
5. 節省文書費用（例如郵寄費用、檔案管理費用等）
6. 專利處理作業透明化

EPOLINE 作業程序

1. 使用歐洲專利局提供之軟體（EASY）輸入書目資料
2. 將以文書處理軟體製作或掃描之說明書附加上去
3. 以 EASY（加密及線上送出）提出申請案
4. 在連線時申請案經電子簽章並加密
5. 由歐洲專利局送出電子收據予寄出者

當申請案提出後，檢索審查委員將作前案檢索，列出所有與申請案有關之文件於檢索報告書上，申請人可以閱讀檢索報告書後決定是否繼續專利申請程序，檢索報告書存在一個資料庫中，目前申請人收到的是紙本，但經由 EPOLINE 申請人可獲得：

1. 檢索報告書
2. 相關文獻
3. 專利家族資料

4. 修正之摘要

EPOLINE 對於歐洲專利局及申請人不僅節省了金錢, 也節省了許多行政作業，加速了專利核准時程。其技術十分進步，足堪本局借鏡。

伍、 esp@cenet

專利涵蓋範圍廣泛、 包羅萬象的科學與技術資訊，多年以來，要檢索並獲得所需之專利資料常須花費許多時間及金錢，因此在 1970 年代，因應需求，出現了許多收費的線上專利資料庫，例如 PATOLIS、WPI、INPADOC 等等，在 1980 年代晚期，有光碟資料庫，時至今日，由於資訊科技的重大突破，網際網路的蓬勃發展，影響無遠弗屆，成為提供專利資訊之最佳途徑，美國專利商標局於 1995 年開始透過網際網路提供 20 年之專利文獻；日本特許廳亦於 1996 年將包括英文摘要之專利文獻提供於網際網路；IBM “Internet Patent Server” 則於 1996 年提供美國專利文獻於網際網路，專利資訊之提供日趨開放。

歐洲專利局遂於1997年中開始計畫建置一個資料庫，整合的其所有的專利資料來源。主要目的在提供全球一個免費、簡單、易用的專利資訊資料庫。由計畫完成至實現僅短短六個月，於1998年十月正式啟用esp@cenet。

esp@cenet是目前世界上最大的免費網際網路專利資料庫，包含了超過三千一百萬件文獻，其介面語文為所有會員國之語文。是歐洲專利局推展其專利資訊的最新發展方向，不僅提

供歐洲專利資訊，更提供許多其他國家之專利資料，甚至回溯至 1900 年的專利文獻。它提供：

- 所有供歐洲專利審查委員參考之專利文獻
- 其十九個成員國最近之專利申請案，〈至少近二年〉
- 歐洲專利申請案及 WIPO〈World Intellectual Property Organization〉資料

設定對象

esp@cenet的目標是提供易於使用之系統介面，使目前未能接觸專利資訊的廣大一般使用者〈非專家〉可輕易獲得專利資訊，因此建議任何人均可將esp@cenet作為檢索專利之開始步驟。

esp@cenet資料特點

- 以歐洲專利局之資料來源為基礎
  - 為專利審查官所建之資料庫之再使用
  - BNS影像資料庫之再使用
- 簡易的web based 介面
- 世界專利資料(超過50個國家)
  - 甚至回溯至1900之前的資料
  - 約4千萬筆書目資料
- 可檢索之書目及摘要

- 所有的專利名稱及摘要均為英文

- 顯示詳細的資料

- 說明書首頁影像、圖及全文

- 將近三千筆專利文獻、一億五千萬頁（A4尺寸）以上影像資料

- 範圍廣泛、更新快速

Esp@cenet資料庫涵蓋範圍

國家或組織	國家代碼	書目資料起始年份	說明書影像起始年份
非洲智慧財產權組織	OA	1966	自始
非洲地區工業財產權組織	AP	全部	自始
阿根廷	AR	1973	無
澳洲	AU	1973	無
奧地利	AT	1975	1920
比利時	BE	1964	1920
巴西	BR	1973	無
保加利亞	BG	1973	無
加拿大	CA	1970	Unique
中國大陸	CN	1985	無
克羅埃西亞	HR	1994	無
古巴	CU	1974	無
塞普勒斯	CY	1975	無
捷克	CZ	1993	無
捷克	CS	1973	無
丹麥	DK	1968	1920
埃及	EG	1976	無
歐亞專利組織	EA	1996	無
歐洲專利局	EP	1978	1978

芬蘭	FI	1968	自始
法國	FR	1968	1920
東德	DD	1973	Yes
德國	DE	1967	1920
英國	GB	1969	1920
希臘	GR	1977	無
香港	HK	1976	無
匈牙利	HU	1994	無
印度	IN	1975	無
愛爾蘭	IE	1973	1996
以色列	IL	1968	無
義大利	IT	1973	1978
日本	JP	1973	1980
肯亞	KE	1975	無
韓國	KR	1978	無
拉脫維亞	LV	1994	無
立陶宛	LT	1994	無
盧森堡	LU	1960	1945
馬拉威	MW	1973	無
墨西哥	MX	1981	無
摩爾多瓦共和國	MD	1994	無
摩納哥	MC	1975	自始
蒙古	MN	1972	無
荷蘭	NL	1964	自始
紐西蘭	NZ	1979	無
挪威	NO	1968	無
菲律賓	PH	1975	無
波蘭	PL	1973	無
葡萄牙	PT	1976	1980
羅馬尼亞	RO	1973	無
俄羅斯	RU/SU	1972	無

斯洛伐克	SK	1993	無
斯拉維尼亞	SI	1992	無
南非	ZA	1971	無
西班牙	ES	1968	1969
瑞典	SE		
瑞士	CH	1969	1920
土耳其	TR	1963	無
美國	US	1968	1920
越南	VN	1984	無
世界智慧財產權組織	WO	1978	自始
南斯拉夫	YU	1973	無
尚比亞	ZM	1968	無
辛巴威	ZW	1980	無

目前esp@cenet每天被存取超過五萬次（甚至十萬次以上），每天有六萬個不同的使用者（依據IP位址），一星期有二十四萬個不同的使用者，每星期對全世界提供超過一百萬頁以上的專利文獻影像資料，其使用者數量持續大幅成長。

esp@cenet資料庫由於其設定目標為公眾，特別是個人、小型或中型企業希望可以獲得專利資訊者，而不是針對專利專業人員（這類人員有其他資源可供利用），因此在設計上，使用者僅須極少或不需訓練即可執行關鍵字、發明人名稱、申請人名稱及號碼等檢索。獲得檢索結果時更可進一步以超連接方式獲得全文、影像及專利說明書。但是該資料庫未提供複雜的檢索運算功能，如截字檢索、鄰接檢索等，對於專利專業人員，功能似嫌不足。

資料庫中可檢索及顯示之欄位如下所示：

公開號　　Publication Number
優先權日　　Priority Date(s)
公開日　　Publication Date
申請人名稱　Applicant Name(s)
申請號　　Application Number
發明人名稱　Inventor Name(s)
申請日期　　Application Date
國際專利分類 IPC Classification
優先權號　　Priority Number(s)
專利名稱（英文）　Title Text (in English)
摘要（英文）　　Abstract Text (in English)

檢索到的結果先顯示公開號及專利名稱，並可超連結其書目資料，依據資料公開時間之不同，可以提供的資料分別為：

- 1970 年以後公開之資料，其”代表性文獻”將顯示書目資料、英文摘要，並可超連結至其首頁影像、請求專利範圍全文。
- 1970 年以前公開之資料，一般而言可以提供摘要及全文，但說明書將以影像檔提供。
- 1920 年以前公開之資料僅提供書目資料。

在目前全文影像是以一頁一頁的方式提供，但歐洲專利局計畫於近期內提供整份文件傳送之功能，整批的資料亦可由各國專利局提供。

這些專利文獻將可在線上直接獲得，自下指令至獲得文獻之時間差將很短。

陸、結論與建議

1. 加強輔導民眾檢索及利用專利資訊：

專利資料為極佳的產業技術研究發展參考資料，台灣在傳統產業競爭力已日益減弱，輔助業者善加利用專利資料, 加強研究開發, 將產業轉型為高科技產業更形重要，又善用資料除可避免不必要的研發投資，節省寶貴的人力及經費，更可開發更尖端技術，獲享更高利益；亦可避免專利侵權糾紛，避免付出高額侵權賠償金或權利金。然國內業界目前專利資料利用情形不甚普遍，對專利資訊之利用仍有待加強，身為專利主管機關，本局責無旁貸，當再加強專利資料檢索與應用之宣導。

2. 正視專利審查人員不足之問題：

歐洲專利局職員人數逾五千人，但仍感人力不足而持續增加員額中, 以西元 2000 年為例即進用人員達數百人，顯見歐洲專利局深知專利於產業經濟之重要性，且專利申請案件數量仍將維持逐年上昇之趨勢，而專利前案檢索人員及審查人員非經多年之訓練, 指導, 無法有良好、令人信服的檢索及審查品質；我國專利申請案件數量亦

逐年大幅成長，但檢索及審查人員數量卻未相對增加，工作負荷過重，恐難再提昇專利審查品質。

3. 儘速引進 MIMOSA 軟體技術：

近年來資訊技術發展日新月異，且通訊網路的發達，更使得資訊傳遞沒有時間及距離的差別，在我國由於英譯專利資料庫市場有限，且專利資料內容多屬於具新穎性的高科技知識，查閱者多為研究發展人員，其要求品質較高、所需檢索功能較強。檢索軟體由我國自行開發恐不敷成本，又品質是否符合業界需求，恐有疑慮，而 MIMOSA 軟體目前擁有龐大的使用者群（已有五十餘國採用 MIMOSA 軟體發行資料庫），其檢索功能極強，且仍將因應使用者需求及技術成長而不斷更新，愈趨強大及完備。因此，引進 MIMOSA 軟體，除可降低開發成本，縮短建置時間，更可避免在資料庫流通方面，其檢索介面未能與他國相容之疑慮。

4. 儘速建立良好專利資訊檢索系統：

良好的前案檢索資料庫，為提昇審查品質之必要條件，完善的自動化流程，更可加快審查程序。若資訊系統於規畫之初，即將輔助審查納為主要需求，將可大幅縮短審查時程，並提昇審查品質

5. 資料庫資料的標準化

在資訊的流通上，除語文的限制外，資料的格式亦為一重要關鍵，各國的資料格式若不一致，將造成資料庫整合及發展上的困難，世界智慧財產權組織有鑑於此，已依據專利資料之特性，制定了一系列之標準。本局資料庫製作若能遵循該等先進國家採行之標準，不僅能提昇本局資料庫之品質及檢索功能，更有利於本局與其他國家資料交換。

6. 加強國際合作：

拓展國際外交的最好方法之一就是多方參與國際組織之交流活動，例如 WTO、WIPO、EPO(歐洲專利局)等組織，雖然我國並非其正式會員，但仍可藉由彼此間的合作關係，獲得技術資訊之實質利益，並達成促進交流、躋身國際舞台之目的。目前與 EPO 最重要的合作項目是英文專利資料的交換，經由合作關係，對於日後我國專利資料庫建置政策及發展方向之擬定極具參考價值。加強國際合作，不但能引進國外新知，增進科技發達，還可提昇我國之國際地位。

附錄：

第1章　簡介

MIMOSA 的名稱是由MIxed MOde Software Applications(混合模式應用軟體)字頭縮寫所組成，主要包含有兩大功能：

1. 將專利或商標資料集合製作於媒體中，如：光碟片

2. 資料檢索、擷取、查詢、顯示及列印

此軟體是由日本特許廳、美國專利商標局及歐洲專利局三方合作開發而成，其目的如下：

- 處理智慧財產權資料：主要指專利和商標資料
- 混合模式資料處理：文字用SGML格式，影像用Gr 4 bitmaps(點陣式)TIFF檔案格式
- 在光碟或其他媒體(例如：硬碟)中製作多元化的專利集合資料(例如：首頁資料First Page，全文資料Full Documents，主要鍵值索引資料(Master Indexes)
- 在微軟視窗(MS Windows)作業環境下可檢索、擷取、顯示及列印資料
- 檢索、擷取、顯示及列印影像資料(例如：ESPACE系列的光碟產品)

本軟體工具包括兩個部分：

- **製作軟體 / 資料準備(Authoring / Data Preparation)**系統

  開發製作專利或商標資料符合MIMOSA特定格式之上產品
- **資料檢索、擷取、顯示及列印**系統

  處理經由上項功能所產生的資料，以及使用其他軟體製作之專利資料

本操作手冊章節內容介紹如下：

第二章：資料　　建立新的資料集合

第三章：製作軟體	如何使用製作軟體
第四章：資料準備	資料準備工作
第五章：建立資料索引	產生索引資料
第六章：線上工具	輸出樣式之產生及修改COLLECT.INI
第七章：參考資料	查詢詳細說明資料
第八章：附錄	常用名詞索引

本操作手冊皆是以EPO-A之出版產品專利首頁資料為範例，原因在於所有索引欄位顯示及輸出格式設定等功能都可以涵蓋在內，有關手冊中的資料索引及輸出格式設定，則以EP-WO產品為範例。

本操作手冊中在介紹**製作軟體 / 資料準備(Authoring / Data Preparation)**系統之作業環境是建置於UNIX-HP™系統下，除此之外，同樣也適用於UNIX-SUN™系統環境，但對於DOS作業系統則有些許差異及限制(註：現行已不使用)，此部分將於後述再討論。

第2章　資料

2.1 沿革

專利及商標電子資料皆源自於紙本，後因電腦設備的備普及，發展出各式各樣的儲存媒體，從大型主機資料庫系統，到商業用資料庫，甚至於個人電腦及光碟片。使用光碟片儲存索引書目(INID Codes)和影像資料資料，具有可攜性，因此在國際各國專利機構間愈來愈普遍，但它的缺點是影像資料需要佔用大量的儲存空間，以及檢索時會受到種種限制(僅限於摘要等欄位的書目文字資料)。

有鑑於改進上述缺點，共同決議出結果為，在處理資料方面，文件中所有的書目及文字資料一律採用SGML標準格式處理，但為結合所對應的非文字圖形檔(可插列於文件中)，因而考量混合資料模式(Mixed Mode format)，應用於資料相互傳遞上會較為理想，改善後的優點如下：

- 使用混合資料模式取代全影像文件節省了大量的媒體儲存空間
- 易於建立資料索引，SGML文字格式使用ASCII碼，整篇文字中更具鄰字檢索及前後

  截字檢索功能

2.2 功能簡介

MIMOSA之**製作軟體/資料準備**軟體工具可在光碟片或硬碟機中建立資料集合，雖然本系統是針對專利商標資料而設計，但也可應用在其他具有類似結構的資料上，換言之，科技文件資料亦可適用，但本系統並不是為製作如書籍閱讀方式的資料庫。

對於如何建立新的資料集合，在第三章中會有詳細介紹。

在MIMOSA的研發期間，共建立了六個光碟雛型測試版本，證實了各種資料集合只需透過組態檔中的變數設定(**第三章*configuration files***)都可適用，各類資料應用如下：

- 使用混合資料模式處理專利說明書全文資料(僅為EPO專利文件)

- 從EPO的EPOQUE資料庫中篩選出專利說明書首頁資料，其中結合書目資料、摘要以

  及一個”附貼式”(與文字併存)的圖形
- 專利說明書全影像資料
- 商標資料庫(僅美國專利商標局USPTO之資料)
- PAJ 資料庫(日本專利說明書首頁之英文資料)
- Cassis 資料庫(僅限書目資料)

現階段之MIMOSA產品計有：

- PAJ光碟片，內容包括日本專利說明書首頁英文資料
- EP-A和WO(世界)專利說明書首頁，取代原ESPACE FIRST產品之影像資料
- DG-1 影像資料，1978以前特定類別之文件

預計將發行之產品有：

- EP-A和EP-B混合模式之全文專利資料
- 專利說明書首頁資料之光碟片
- 各國專利說明書首頁資料

(註：上述資料現皆已發行)

第3章 製作軟體

3.1 製作與資料準備

本軟體—**製作 / 資料準備（*Authoring / Data Preparation System*）**—的標的是針對光碟片及其他儲存媒體(如：硬碟、DVD)。**製作**部分是用於建立新的資料集合及其設定；**資料準備**部分則是用於將已經存在的資料集合，產生索引等相關檔案，在本章中將討論到

建立新資料集合之各項設定，包括：

- 製作一片新的資料集合的基本步驟
- 會用到那些軟體工具
- 如何使用製作軟體

3.2 基本步驟

在設定製作軟體操作環境(章節3.3)之前，對於新的資料集合，有幾個基本項目必須先行確認：

3.2.1. 資料內容

新資料集合中該包含什麼內容？是完整還是部分(如：專利首頁資料、要不要包括專利申請範圍)的專利資料？另一個重點是資料範圍，是否選擇一個特定類別的資料？還是以時間分割，選擇一段特定時間內的資料？或是以國家作為資料界定範圍等。

(註：文件中內容若來自其他來源，不列入本手冊討論範圍)

3.2.2 資料檢索 / 擷取

為資料檢索，必須指定那些是用於索引排序的鍵值基本欄位，(例如書目資料中的專利號、分類號、優先權資料、發明人、申請人等欄位)，及排序方法(是用人名的全名，還是姓名分開，各字獨立排序等等)。在全文檢索時，還要考量到子欄位、鄰字檢索、前後截字檢索等條件，詳細介紹請參考 ***3.3.2 SGML2GTI***。

3.2.3 資料輸出呈現的版面配置

使用混合模式，還有一個很重要的設定就是，資料在輸出時，各欄位、影像等資料所呈現的版面要如何配置，例如書目資料及專利圖形可併頁排置，每頁資料在輸出時，要不要分割為左右版面？用什麼字型？用多大的字體？每頁資料要容納幾行字？那些資料可以合併在一起顯示？那些要分別排置？這些都該預先就要設計得清清楚楚，並準備好範本，對資料輸出格式而言，這些是最重要的工作(請參閱9.9.9輸出格式)。下例即是EP專利文件的說明書首頁的版面呈現。

Sample first page data

在只顯示專利書目資料的情況下，又該如何安排版面？這是對SGML2GTI 的基本規範需求

Sample bibliographic notice

各欄位名稱該如何標示？除了專利號等辨識文件的唯一欄位外，其他應該標註什麼欄位？標題或是分類？這些設定也都在SGML2GTI中會規範到。

3.2.4 輸入資料的標準格式

MIMOSA使用的輸入資料是採混合模式的資料格式，也就是說文字用SGML格式，影像用Gr4 bitmapped(點陣式)格式，有關格式定義請參考WIPO第30、33及35號標準(ST.30, ST.33 and ST.35)

3.2.4.1 SGML

在MIMOSA系統中對於書目及一般文字資料使用SGML格式，這不但是建立索引的基本要求，同時也會應用在資料螢幕顯示及列印等輸出時的格式設定上，有時還用在檢查資料的完整性，因此，對輸入資料而言，文件型態定義(DTD, Document Type Definition)一定要先予確認。在文件型態定義中，會說明到那些是SGML格式所使用的標記符號(Tag)，應該放在文件中的什麼位置。專利資料部分大多在WIPO第32號(WIPO ST.32)及其相關標準中都有定義，WIPO ST.32及DTD請詳參附錄(12)

The SGML coded bibliographic data of the sample page

3.2.4.2 Gr4 影像檔

本軟體現行支援的影像檔是用Gr4 bitmaps(點陣式)規格，可以是前附256位元標題說明資料辨識內容(參照WIPO ST.33或ST.35標準)的格式，或是以TIFF格式標記(Tag)所有資料辨識內容。有關標題及TIFF標記的說明，請參閱附錄(17)。

3.2.4.3 檔案格式

SGML格式文字資料和Gr4圖形檔可分別儲存於不同的檔案中，或是併存在同一個檔裡。檔案分開儲存是基於原WIPO第30號標準(文字轉換SGML格式)，和原第33號標準(Gr4圖形檔，前附256位元的標題註記)，但前述兩項標準現在已經被第35號標準混合式資料模式所取代，也就是把SGML文字格式和Gr4圖形格式合併放在同一個檔案中。SGML格式的文字可使用ASCII碼或EBCDIC碼，圖形可以是前附256位元標題註記或TIFF格式，我們建議使用者儘量採用本項標準，取代前述之原第30和33號標準，詳細說明請參閱附錄(14)。

3.2.5 輸入資料的儲存媒體

系統在輸入來源資料時，亦可透過數種儲存媒體來讀取資料，如1/2”磁帶、IBM卡匣(cartridge)、DAT磁帶，或是經由其他網路可直接拷貝進系統的媒體，詳情請參閱4.1.1。

3.3 系統工具

根據前述之資料輸出及輸入規範，即可設定軟體之作業環境，在程式之外，製作軟體亦需作必要之變數(Variables)設定，系統就是依照這些設定的變數值運作，因此這些變數可稱之為系統工具(Tools)，設定好的變數將儲存在組態設定檔(Configuration Files)及系統相關支援的檔案(Support Files)中，製作軟體作業環境中的系統工具一般是用於光碟片製作，於後***第4章 資料準備***中還有更詳盡的介紹。

在***章節3.3***中將對變數的內容逐一說明，在***3.4 如何使用製作軟體***中，將詳細描述使用

GENCD功能設定作業環境的執行程序。

3.3.1 SGML格式轉換

SGML是針對輸入資料的格式標準，這些資料在送入系統作業前，必須先通過語法完整性和正確性的檢查。

➲ ➲ 完整性：檢查標記符號是否有遺漏？

➲ ➲ 正確性：標記名稱是否正確？是否置於正確的位置(合於文字語法結構)？特殊符號字

元(非ASCII)表示方法是否正確？等等

這些都是透過SGML語法分析程式(Parser)作檢查的，檢項必須完全合格通過程式的文件型態定義(Document Type Definition，DTD)才算有效，DTD是一種正式的文字標記技術，因此必須兼顧完整性和正確性，詳情請參附錄(19)。

DTD通常都是已經存在的，而且也屬於標準需求的一部分，對專利資料而言，都定義於國際智慧財產權組織第32號標準中(WIPO ST.32，附錄(12))。當然，由於各國專利制度不盡相同，在標準中若有未列及的部分，使用者亦可比照語法格式自行定義，這種情形也就是指定義一個特殊的DTD格式。

另一個要檢查的重點是非ASCII碼的特殊字元，每個符號皆有其特定的表示方法，並要符合DTD定義，因此，在DTD中應該提供一個相對應的查核檔案，紀錄特殊字元該如何表示，詳情請參附錄(12)和(13)，DTD與其對應的查核檔案必須儲存在**資料準備系統(Data Preparation system)**可以辨識的目錄之下，這些內容在**GENCD: 1.3.3 設定環境參數**的章節中再討論。

3.3.2 SGML2GTI

資料集合中有些欄位是可以建立排序索引運用於檢索上，因此在軟體中必須先定義那些欄位資料要建立索引？索引鍵值為何？如何命名？以及提供的檢索功能有那些？例如鄰字檢索、截字檢索、日期檢索，字串檢索等。

此外，還應該要確認書目資料檢索時的限制條件，最大檢索極限的界定，以及定義一份完整的資料該有那些內容。上述條件皆定義於SGML2GTI組態設定檔(configuration file)中。

所有定義的內容全根據於SGML標記以及標記的正確屬性，SGML2GTI程式也是依據SGML2GTI定義(資料索引時的特殊組態設定)來撰寫。

在上例中的主要指令定義如下：

Hitlist definition

HITLISTDEF = <B190>" "<B110>" "<B130>" "<B140>

In the **HITLISTDEF** keyword the tags of the fields that must be stored in the hitlist and how the fields are separated (or concatenated). In this example the fields are separated by a blank. The sequence of the tags is also used to specify the default order in which the hitlist will be displayed. (In this case the order of Office - Publication number - Document Kind – Publication date).

Notice definition

NOTICEDEF = ("PN : "<B190>" "<B110>" "<B130>" "<B140>,

SFTAGSIGNORE=<TIME>

CRITNAME = "Publication Number display"

CRITABBR = "PN")

Each keyword **NOTICEDEF** contains a specification of a line in the bibliographic notice view, in this case the fixed text *PN:* followed by the contents of the tags <B190>, <B110>, <B130> and <B140>. The fields are separated by a blank.

With the sub keyword **SFTAGSIGNORE** one can specify which sub tags in the selected tags must be ignored. In this case the tag <*TIME*> and its contents are not displayed in the notice line. From the example it can be seen that sometimes many sub tags are ignored, e.g. for <B711> *Applican*t.

With the keywords **CRITNAME** and **CRITABBR** the full name and the abbreviated name for the notice line are specified. Se also below under Index definition.

Typography

typography = (<above>,), (<altmath>,), (<ano>,),

(<Atl>," "), (<AUTHOR>,$q), (<b>,), (<bai>,),

(<BNUM>, " "), (<BOOKId>, $q)

With **typography** it is possible to add some (very) restricted layout to the notice (the full Mixed Mode displays the bibliographic data “as printed”). E.g. insertion of a line feed, or an extra space. The $ commands refer to the GTI commands, see (99) for detailed information.

Document identification

DOCUMENTID = "%2.2<B190>s%-2.2<B130>s %08<B110>ld"

The keyword **DOCUMENTID** specifies how a *unique* identifier can be built; each document must be distinguished from the other documents in the collection. The numbers following the % character indicate the number of characters or digits to be used from the SGML fields and how these are placed in the identification field. In this example the office/country code followed by the publication number and the document kind form the unique identification.

Index definition

INDEXDEF = (TAG2GTI = <B190><B110> <B130>“ “<B110><B130>“ “<B110,

CRITNAME = "Publication Number" ,

CRITABBR = "PN")

The keywords **INDEXDEF** is used for the definition of indexes. This is the most important part of SGML2GTI. Apart from the content, it contains also specifications about how the content must be indexed. The following sub keywords can be used: **TAG2GTI** contains the SGML tags of the fields to be indexed. This can be one or more tags, e.g. a main classification index from the tag <B511> and all classification tags in another index (tags B511, B512, B513, B514). See the example above. When the tags are separated by a blank then the contents of the tags are indexed separately, when the tags are placed adjacent to each other

then the contents of the fields are combined as one index term. In the Publication number example the fields are indexed as: Publishing office - Number – Kind (EP0717989A1), Number - Kind (0717989A1) and number (0717989).

With the keywords **CRITNAME** and **CRITABBR** the full name and the abbreviated name for the index are specified. The names can be the same for the NOTICEDEF and the TAG2GTI.

Other sub keywords are: **SFTAGSIGNOR**E, see above under Notice definition **DATATYP**E, this specifies how the field must be indexed. If the keyword is not present then the indexing default is used: the content is indexed as one field. Some datatypes are:

DA the field contains date information

ST8 the field is a classification code that must be indexed according to the WIPO ST.8 standard

LI the complete string is indexed, e.g. from the applicant name LABORATORIOS CUSI, one index term is built: “LABORATORIOS CUSI”,

LIWO the complete string and the separate words must be indexed, e.g. from the applicant name LABORATORIOS CUSI, three index terms are built up: “LABORATORIOS CUSI”, “LABORATORIOS” and “CUSI”.

IN and *INL* indexed as a string of 5 or 9 digits respectively (leading zero’s are added when the input is less then 5 or 9 digits.

Other less used datatypes can be found in (10).

PROXIMITY = Y or N or the word proximity only specifies if the text must be indexed so that proximity searching is possible. When the key word is not present: no proximity is the default.

CONTENT indexing occurs only when the specified tag contains a specific value. E.g.

```
INDEXDEF = (TAG2GTI = <B542>,
                    CONTENT = <B541>="DE",
```

tells the software to index the contents of tag <B542> only when tag <B541> contains the word “DE”

ATTRIBUTE only the SGML tags with the attributes containing the specified value are indexed.

INDEXDEF = (TAG2GTI = <SDOAB>,

ATTRIBUTE = LA="E",

This tells the software that only English abstracts are indexed.

TRUNCATE specifies the type of truncation, in cases of *L* or *B* a mirror index is created. E.g.

INDEXDEF = (TAG2GTI = <B542>,

CONTENT = <B541>="DE",

TRUNCATE = B,

tells that both left and right truncation must be possible. In our example:

<B541> DE </B541> <B542> VERFAHREN ZUR UMHÜLLUNG VON

NANOPARTIKELN ODER -TRÖPFCHEN </B542>

the words are indexed as follows (stopwords are ignored):

VERFAHREN

NERHAFREV

UMHULLUNG

GNULLUHMU

NANOPARTIKELN

NLEKITRAPONAN

TROPFCHEN

NEHCFPORT

STOPWORDS this keyword contains the name of the file containing the stopwords being used for this index. E.g.

INDEXDEF = (TAG2GTI = <B542>,

CONTENT = <B541>="DE",

TRUNCATE = B,

STOPWORDS = “DEUSTOP.LST”

indicates that DEUSTOP.LST contains the stopwords for this index and shall contain e.g. the words *zur, von, oder* from our example.

DISPLAY this keyword tells the software that not only an index must be built but also that the contents must be displayed in the notice as an addition to the notice definitions above.

3.3.3 影像辨識

一份專利文件，除文字之外，幾乎都含有影像資料，所謂的影像資料包括示意圖，掃瞄存檔的文件，或是一個插在文字中的各種圖式，這些圖式無法使用SGML格式來表示，例如化學結構式、複雜的表格或公式、及機械製圖等。

對於系統軟體而言，必須要知道那個影像屬於那一份(專利)文件，應該要放在文件中的那個位置，這些辨識動作，是透過一連串的指令執行：

➲ ➲ 使用TIFF格式影像檔，必須另建立一個連結資訊檔(Pointer File)，在這個檔案中紀錄

每一個影像的內容說明(即混合資料模式或是ST.33中所謂之標題內容)及其存放位置

➲ ➲ 同樣地，在SGML文字格式的檔案中，也必須建立影像識別資訊，才能讓軟體辨別那

一個影像屬於那一份(專利)文件，這部分是屬於資料結合的範圍(請參閱**資料準備**)

在稍前SGML2GTI中已討論過，每一份(專利)文件，都應有唯一的辨識定義，而影像該放在文件中的什麼位置，都是使用SGML中的標記指令來表示，因此在軟體中需先定義那些標記指令是用於影像，以及標記內容的位置，才能找到正確對應的圖形，在連結資訊檔中要建立一個指向影像的連結指標，讓系統知道影像檔存放在那裡，範例中，影像標記指令為**<EMI>**，識別影像屬性的名稱為**ID**，後續**GENCD: 設定環境參數**的章節中，還有詳盡的介紹(請參閱3.4.6).

3.3.4 光碟片

對MIMOSA系統而言，現行的標的媒體為光碟片(CD-ROM) ，雖然亦可使用硬碟或DVD，但這不列在本操作手冊討論範圍內。光碟片的規格必須依循*ISO9660號標準*(附錄(20))，於此項標準中對於光碟(如專利資料、百科全書、套裝軟體等)之外在標示，卷期、內容概述及版權聲明皆有最低限度的明文規定，同時對於使用之欄位、檔案及目錄名稱

之命名規則亦有說明，一般名稱只能使用0-9數字、A-Z大寫英文字母及底線_等字元。基於此項標準，就必須列出光碟內容及其說明方式的規範，詳細規範可參考ISO9660.CFG檔案，ISO9660.CFG檔提供了檢查現行檔案及是否合於命名規則的功能，並會隨時依需求更新，詳情請參(附錄9及附錄20)，下列為檔案範例：

ISO9660.CFG (部分摘錄)

在此範例中，下列所示之主要定義都非常地明確易識：

V.Soft　# S0.1

V.Disk　# D1.0

V.Soft 和 **V.Disk** 分別代表著軟體及光碟片的版本，無論任何軟體，應有義務標示這些訊息，但MIMOSA軟體現行不是採用此種標示方法。

欄位的左方為固定字元，資料欄位訊息說明將會儲存在光碟之**主要內含描述(*Primary Volume Descriptor，PVD)***中。

Volume # EPFD19%year%%ncd%

Volume(卷期) 是用來指明光碟卷期名稱，這個名稱必須縮減在3至4個字元中，之後的字元用於標示出版序號，相關訊息儲會存於PVD中，這幾個縮寫字元非常重要，因為MIMOSA軟體就是根據這幾個字元辨識出光碟中所含的資料，然後才能正確地對映出輸出格式及索引排序等相關資訊，本項資訊皆儲存在資料集合的檔案中，請參閱6.2。出版訊息最佳的表示方法是透過 **%year%%ncd%** 參數設定，(西元)年代用後二位數表示，前二位固定為19(註：此處為19XX之年代表示)，這年代後二碼，即可判別光碟之唯一性，在本例中ISO9660.CFG檔後續並無任何更新資訊，變數說明請參**GENCD:設定環境參數**(3.4.6)

VolumeSet # EP_FULL_PAT

Volume Set Name (光碟片名稱) 儲存在PVD中，當使用光碟片時，名稱會自動顯示，名稱長度限制在127個字元內，但為提高可讀性，應儘量縮減並使用具有意義的名稱。

Application # MIMOSA V3.0+

Application (應用軟體) 包含了軟體的名稱(Mimosa)及最初之版本代號，以供讀者諮詢使用

Publisher # THE EUROPEAN PATENT OFFICE

Publisher (出版者) 資料之製作者

File # \CONTENTS # %datasgml%\abstract # ABS

File # \COPYRGHT # %datasgml%\copyrght # COP

File # \BIBLIOGR # %datasgml%\bibliogr # TXT

File # \MIMOSA.ROM # %database%\mimosa.rom # ROM HID (MIXMOD)

File # \IMAGE # %database%\image # TXT HID (MIXMOD)

File (or **Fichier) (檔名)** 上列檔案描述都必須存在光碟中，依照所指定的目錄路徑，便可開啟硬碟中的檔案，並註明其檔案形態，內容說明如下：

➲ ➲ 在第一個和第二個 # 符號間的文字是說明, 檔案在光碟片中的儲存路徑, 如非在根

目錄, 即是在 \ 符號後的目錄路徑名稱下, \ (反斜線)是目錄層次的區隔符號。

註：目錄名稱及檔案名稱都必須為大寫字元

➲ ➲ 在第二個和第三個 # 間的文字是用於說明硬碟中的檔案儲存位置, 這些變數可設為

固定, 如此就不需要時常作修改, 變數名稱於**GENCD: 設定環境參數**(3.4.6)中討論

➲ ➲ 第三個 # 後的文字是說明檔案形態, 常見檔案可分為下列幾種：

COP	版權聲明檔, 此檔儲存於光碟中, 且一定需在根目錄下, 檔名儲存在**主要內容描述**的*copyright file name(版權檔名)*欄位中
ABS	附有光碟內容摘要說明, 此檔儲存於光碟中, 且一定需在根目錄下, 檔名儲存在**主要內容描述**的*abstract file name (摘要檔名)*欄位中
ROM	表示檔案包含索引排序及為GTI格式, 此種檔案用於MIMOSA軟體之資料檢索
TXT	文字檔, 包括二位元檔, 幾乎任何檔案都可以是文字檔
EXE	執行檔, TXT和EXE檔皆沒有特殊之其他說明, 執行檔亦可儲存為文字檔(TXT)格式
HID	檔案屬性為隱藏, 在DOS系統下, 檔名不會顯示
(MIXMOD)	表示檔案中包含混合資料模式的資料

此處可指定多種不同型態的檔案, 例如軟體安裝檔案如下：

File # \%DTD% # %cfg%\%dtd% # TXT

除上述之外, 光碟中還有其他的檔案形態含有不同的內容, 例如：DTD是指SGML格式設定, DTD的檔名源自於系統環境變數(GENCD), 透過%DTD%變數中, 檔案名稱會自動從指定的目錄下, 載入%cfg%變數中。

其他尚有：

- MIMOSA軟體之安裝程式，在使用軟體之前必須先完成安裝程序
- 新資料集合之輸出版面格式及**COLLECT.INI**檔的其他資訊，請參**第六章 線上工具**
- 操作手冊，管理手冊等
- 畫面展示，使用PowerPoint圖示MIMOSA軟體各個部分如何運作以及其他功能說明

由於在軟體安裝時必須加入大量的檔名，這些檔名也可以在*GENCD環境變數設定* ***cd_files***主要目錄中指定其路徑位置，在產生*GENCD ISO9660*的過程中，檔案ISO9660.CFG會自動拷貝到主目錄之下，以供前述所有檔案使用，子目錄下也會建立這些檔名的清單。在光碟片中，上述儲存於主目錄下的檔案會置於根目錄下，其他原本在子目錄下的檔案，則依序置於同(大寫)名稱的子目錄下。

3.3.5 光碟中與標準有關的檔案

在章節 **3.3.4 光碟片** 中提到二個屬於ISO9660標準的檔案，*Abstract* or *Contents內容檔* 及*Copyright版權檔,* 前者是說明光碟片的內容以及(以出版者認定)對讀者最重要的注意事項；而後者是版權相關訊息。

另外還有一個*Bibliographic書目資料檔,* 說明MIMOSA軟體中資料檢索名稱及其使用方法。

上述這些檔案都是ASCII文字檔。

3.4 如何使用製作軟體

在之前的章節中介紹了環境中各種用到的工具程式，本章節中將要說明如何使用**製作軟體/資料準備(Authoring / Data Preparation)** 設立一個新資料集合的所需環境。

3.4.1 檔案結構

光碟片中的資料運作，如輸入、資料暫存、輸出等動作，需要用到很多磁碟空間，因此為處理大量的資料，必須在磁碟中設定一個工作目錄區及配置足夠空間，這個步驟是為避免因空間不夠導致覆蓋造成資料流失，可能的話，最好再利用另一個磁碟，建立分散式目錄結構，下例中即是開啟兩個工作目錄(Unix系統下的目錄名稱為向左靠齊)：

- **data**目錄 包括input, intermediate, report files和output四個子目錄，在光碟製作完成且備份後，可將子目錄刪除。
- **software**目錄 包括***製作軟體/資料準備***軟體及所有的組態設定檔和其支援檔案(系統工具)，這個目錄必須經常備份且要永遠存在。

下列為在本手冊範例中會討論到的目錄結構示意圖：

Fig 1: Main directory structure

3.4.1.1 Authoring 目錄

在 *authoring* 目錄下之各子目錄中，包括其程式和相關組態設定的所有檔案，其子目錄結構說明如下：

Fig 2: Software directory structure

gencd	此目錄下包含GENCD工具程式及相關的組態設定檔案(其附加檔名為.GCD)。
epfdcfg	此目錄下包含EPFD資料之組態設定及相關檔案，其他資料也會有類似的子目錄在這個目錄下，例如有DTD、其相關檔案、和資料特定程序用到的檔案(AWK程序)、iso9660.cfg和sgml2gti.cfg(最後二個檔案將於3.3.2及3.3.4中說明)。
tools	所有的執行檔都存在這個目錄之下。.
idc	此目錄下包含用於建立光碟中所有索引的組態設定及相關檔案。
cdtrav	這三項索引內容為由Jouve(此為開發MIMOSA軟體之公司名稱)GTI套裝軟體中產生之索引及iso所有權軟之資料。本軟體亦適用於其他非MIMOSA的系統。
mefgti	
iso9660	

3.4.1.2 Data 目錄

在叫做***data***的目錄下，在每期光碟執行資料準備的動作時，都會建立其下的子目錄，此例為*epfd*(EPO專利說明書全文)，*97001*(1997年的第一期資料)，對於資料輸入、關啟Tiff影像檔、SGML檔案(包括報表檔及處理過程中產生的暫存檔)，以及產生的輸出檔案(索引、混合式資料庫、光碟影像)等，都會建立相關的工作子目錄。

註：在工作成功地執行完畢後，這些目錄即可刪除，系統亦會視需要情形自動備份。

Fig 3: Data directory structure

97001_in　此目錄下包括從磁帶讀入的檔案，檔案結構必是符合第35號標準混合式資料模式且附加檔名為**.mx**m，如是採用第30和33號舊標準，內容則會稍有不同。

97001_sg　此目錄下包括符合SGML格式之文字檔，由混合模式之輸入資料中所讀出(即97001_in檔). 但同時有些過渡檔、統計檔及報表檔也會儲存在這個目錄下，詳情請參閱第三章。經由不同的附加檔名，可以辨別出為何種型態的檔案，檔案型態說明如下：

sgm: 各式SGML檔案

spy: 格式偵察報表檔

sta: 含統計資訊的檔案

err: 有問題的文件或影像

chk: 總合數檢查檔

ptr: 指標檔，用以指引路徑名稱及文字的起始位置和長度

gti: 資料建立索引時的輸入或過渡檔案

rej: 資料建立索引時，不符合規範之文件

97001_tf　此目錄下包括TIFF格式的影像檔(符合並經轉換為混合資料模式之資料)及用以指示影像位置和大小的指標檔(.ptr)。

97001rom　儲存最後產生的結果檔案：

mimosa.rom　GTI格式之索引及說明

image　包括混合模式之最終資料

iso	要燒錄在光碟中的影像檔

如何建立上述目錄結構，將於後續討論到*GENC*D時再作說明。

3.4.2 檔案內容編輯

所有在**3.3系統工具**中討論到的檔案都是普通的ASCII檔，因此使用一般的編輯軟體都可以建立或修改檔案內容。

在轉換為SGML格式過程中用到的檔案(DTD以及所屬副檔案)，*Authoring*程式部分並不會針對這些檔案檢查是否存在或有效，因為系統會內定DTD和其所屬副檔案已經存在並已使用，然而若使用一個經過修改的DTD，建議最好先建立一個SGML測試檔，測試一下DTD之正確性及有效性。

SGML2GTI.CFG和**ISO9660.CFG**這二個檔案還需配合其他步驟，於後另有說明。

3.4.3 GENCD

現行的軟體系統是由**資料準備(Data Preparation)**程式來設定整合，而在整體系統下，則是由***GENCD*** (GENerate CD，即謂產生光碟資料片)的步驟開始操作，但也有其他在資料準備的過程中可能會採用設定自動執行而較為簡便的方法來測試，本手冊中則針對GENCD來操作。

GENCD 為一組程式，包括MIMOSA軟體在**製作／資料準備**的過程中所有可能進行的步驟，每個步驟並非每一次都要逐一執行，各執行步驟都可設定在運作狀態或非運作狀態，這些動作執行同昤也會關及輸入、過渡、輸出之變數，及執行結果是否成功的報表。

3.4.4 執行GENCD

GENCD這支程式是為新資料集合設定環境，內容描述如下，而在**資料準備**部分的相關程序則於第四章中再討論。

- ➲ ➲ 切換到含有GENCD執行程式及組態設定檔的目錄下
- ➲ ➲ 在***gencd***之後鍵入所屬之組態檔檔名**all.gcd** (含在於安裝程式之內)，但如果新資

料集

合與之前既存的資料集合結構相似，則附檔名為gcd的檔案可作為一個開始之處(例

efp.gcd 為EP專利首頁資料的gcd檔)

註：新資料集合的組態設定檔必須存於另一個新檔名下，以避免蓋到舊檔資料，如同在本手

冊範例使用epfd.gcd新檔名

螢幕上會出現主要功能選單，使用者可移動反白游標，按下*enter*鍵選擇功能選項。

在這個選單畫面下，可使用上、下方向鍵選擇功能選項，游標所在的選項上會以反白狀態顯示，要跳離選嗶畫面可按數字鍵 3或*esc*鍵，其後的畫面操作方法皆相同。

3.4.5 設定應用參數－Setting application parameters

此步驟是由主要功能選項的畫面點選進入，之後螢幕上會出現下列畫面：

APPLICATION NAME, COMPANY 及 REFERENCE 等欄位上請分別填入正確資訊

註：每一個欄位中填入的變數值都需要按下enter鍵確認，螢幕中央會自動出現一個視窗，等待使用者鍵入資料，鍵入完畢按下enter鍵後，所布欄位內容都會隨之更新。

在本範例中，分別填入的資料如下：

EPFD (application name)；

EPO (company)；

Full documents (reference)

按下esc鍵(或數字3鍵)結束本設定畫面。

3.4.6 設定系統環境變數－Setting environment variable(s)

下一個步驟是設定新資料集合使用到的所有變數值，其畫面如下：

(註：請留意螢幕上方各欄位中資訊已經由前項設定而被更新)

以下為各欄變數設定說明：

app=epfd

app must contain the name for the collection. This name is used to create the main directory name for the application.

dtd=st32v3.dtd

dtd= specifies the name of the DTD to be used for parsing. The file must be stored in the directory $cfg (see below under *cf*g).

year=97

year= It is adviced to use the year in 2 digits so that the generated sub directory names remain 8 positions (or less), this is not required for UNIX systems but this is for compatibility reasons with DOS based systems.The year can be changed during regular production of CD's. It can also be used in the volume id of the CD (see above).

ncd=001

ncd= here a sub numbering may be used to create a unique directory structure for a particular production. It can also be used in the volume id of the CD. The number must be changed for every production in order to create a unique volume id (in combination with “year” and “app”).

ST33DATA=

ST33DATA= options are: ***YES*** or ***blan***k, this tells that the 'old' ST33 format is used for storage of Bacon type images. It will be used in combination with the next field.

ST30DATA=

ST30DATA= options ***YES*** or ***blan***k, it tells that the file format is the 'old' ST.30 for the SGML coded data. It will be used in combination with the previous field.

Note: The above combinations of separate files in ST30 and ST33 format are becoming

obsolete as it has been agreed to use the Mixed Mode standard (MMMT).

MMMTDATA=YES

MMMTDATA= options ***YES*** or ***blan***k. Used for the new MMMT standard ST.35, for more information about MMMT see (14).

IDXCOUNT=

IDXCOUNT= if set to YES then the indexed items and the counts are also stored in a text file. This for test purposes.

DELETE=YES

DELETE= if set to YES then the intermediate files no longer needed for further processing are deleted. This saves temporary disk space and it is advised to be used in the production environment.

IMAGEFILE=OPEN

IMAGEFILE= this is set to YES if an ***image*** file must be created without open access index, and set to OPEN when an ***image*** file with open access index must be created. Any other text or blanks cause that no ***image*** file is created.

*Note 1: the **image** file is the file containing the text (in SGML coded ASCII) and images (in TIFF format). The **mimosa.rom** file contains the indexes and bibliographic notices both encrypted in GTI format.*

*Note 2: an **open access index** is a file containing the complete directory path to the documents in the image file on the CD-ROM.*

imagetag="RTI EMI"

imagetag= indicates the SGML tag name(s) for the images in the text, e.g.: ***EMI*** for embedded images and, especially for the Japanese collections, ***RTI*** for replacement of text by an image. The name(s) must be enclosed in quotes (“”).

imageid=id

imageid= gives the name of the attribute in the image tag that contains the image identfier.

Note*: from the tags identified as image tags and it's attribute ID a unique key is created for an image* ***within*** *a document. It forms together with the* ***document id*** *a unique key. A possible key is: EPA1 050000000020001 (EP kind A1, number 0500000, image 00020001).*

editor=$EDITOR

editor= usually set to $EDITOR. In this way the standard editor from login will be used but other editors are possible. The specified editor is used for modification of *configuration files* in the step *Data preparation chain* (see below).

drive=/dev/rmt/3m

drive= this is the path and name of the tape device used for the input, e.g. ***/dev/rmt/3m*** for a 4mm DAT tape unit on a HP system.

chaindir=/authoring

chaindir= this is the name of the main directory containing sub directories with software and configuration files (see figure 2). The sub directories will be specified below.

workdir=/data

workdir= this is the name of the main directory containing sub directories with all input, output and intermediate data (see figure 3). The sub directories will be specified below.

datain=$workdir/$app/$year${ncd}_in

datain= the directory path and name where the mixed mode input will be stored. By making use of variables (starting with the $) it is easy to create consistent sub directories for each

issue that is created during regular production. In this example *$workdir is replaced by **/data,** $app by **epfd,** $year by **97,** $ncd by **001** ({} indicate the field name), _in remains unchanged.* The result in this example is:

datain=/data/epfd/97001_i*n.* See figure 3. The same technique is used in the variables below.

datasgml=$workdir/$app/$year${ncd}_sg

datasgml= the directory name for the SGML data and most intermediate files and all statistical information. Resulting name in example: datasgml=/data/epfd/97001_sg.

datatiff=$workdir/$app/$year${ncd}_tf

datatiff= directory for the TIFF files. Resulting name in example: datatiff=/data/epfd/97001_tf.

database=$workdir/$app/$year${ncd}rom

database= directory for the final database and ISO file. Resulting name in example: database=/data/epfd/97001rom.

scratch=$workdir/$app/$year${ncd}tmp

scratch= used for some temporary files. Resulting name in example: scratch=/data/epfd/97001tmp.

exe=$chaindir/tools

exe= path and name of directory where specific Mimosa executable files are stored. This is the sub directory **tool**s. Resulting name in example: exe=/authoring/tools, see figure 2.

Note: other executables are identified in the settings below.

cfg=$chaindir/epfdcfg

cfg= the directory with the configuration files specific for this collection. Resulting name in

example: exe=/authoring/epfdcfg, see figure 2.

ISO9660=$chaindir/iso9660

ISO9660= the directory with the executables for creation of an ISO file. This is the sub directory **iso9660.** Resulting name in example: ISO9660=/authoring/iso9660, see figure 2.

MEFGTI=$chaindir/mefgti

MEFGTI= the directory with the MEF-GTI executables named **mefgti.** Resulting name in example: mefgti=/authoring/mefgti, see figure 2.

Note: MEF-GTI prepares the input for processing by the indexing software (see below).

CDTRAV=$chaindir/cdtrav

CDTRAV= the directory with the executables for indexing named **cdtra**v. Resulting name in example: CDTRAV=/authoring/cdtrav, see figure 2.

cd_files=/mimosa

cd_files= this is the directory from which all files and sub directories are copied onto CD-ROM. The filenames are autmatically added to the fiel *iso9660.cfg* (see 3.3.4).

各項設定完成後，請按下esc鍵或數字3鍵結束，回到上一層畫面。

註：各欄位設定完畢後必須執行**Save configuration file**後才會將設定值存檔。

3.4.7 組態設定存檔

當開啟環境設定時(如前述)，系統會要求鍵入組態設定檔的檔名，並且在使用GENCD的一開始，會顯示這個檔名，檔案最好能配合資料內容命名，例如：epfd.cfg。

3.4.8 資料準備鏈結－Data preparation chain

在主功能選單下選擇***Data preparation chain***，畫面上會出現一個工作表單選項－**Task(s) launching menu**

3.4.9 工作開始－Task launching menu

在這個畫面下，將會啟動二個主要的功能：

- **資料準備(Data Preparation):** 執行一般正規性的產品製作流程，在下一個章節中將有較詳盡的說明。
- **組態檔設定(Configuration files):** 用於設定新的資料集合。

畫面操作之用鍵說明如下(子功能項目亦同)：

➲ ➲ **2:On/Off,** 數字鍵”2”是一個功能啟動與關閉的開關，在上面例中組態檔設定 *Configuration files*以高亮度狀態顯示，代表現正選定執行的功能項，按下數字鍵”2”

後，功能項前會出現一個字母”A”，表示功能啟動；再按一次2鍵，字母”A”消失，表

示功能關閉。

➲ ➲ **3/ESC:Quit**離開現行畫面進階至下一個畫面。

➲ ➲ **Enter:Expand/Collapse** 按下*enter*鍵，會展開所屬的子功能選項，再按一次就會隱藏

收起，在上例中*Configuration files*項目之前顯示了一個加號”+”，表示這個項下還有子

功能項，按下*enter*鍵就可以叫出。

➲ ➲ **9:Exec one** 數字鍵”9”是一個執行鍵，每按一次，會執行一個動作，當選定主功能後

展開子功能選項，即可使用此執行鍵，以高亮度標示的功能項才會被執行。

➲ ➲ **0:Execuate all** 所有被選定的功能項(即之前有標示A的項目)，按下數字”0”鍵，系統

可以一次集體執行。

3.4.10 組態設定檔－Configuration files

將游標反白選項移動到*Configuration files*後按下*enter*鍵，所有工作項目顯示如下：

在這個延伸出的設定畫面，所有的檔案內容都可以編輯，按下數字鍵”2”出現字母”A”就表示已被啟動；每執行一個動作，按一次數字鍵”9”，在環境變數設定*Setting environment variable(s)*時，就可指明使用何種編輯器，然而要編輯這些組態檔案，只要是使用ASCII碼的編輯器，系統都能接受，但必須在系統啟動前就要設定完成，在組態檔的設定畫面

中有些檔案沒有後續的程序，但有些檔案必須依設定值在此畫面中執行一些動作，檔案說明如下：

Sgml2gti Configuration file Modification	從環境變數設定目錄中選擇“*cfg=“*，設定後的內容會儲存在**sgml2gti.cfg**檔案中。 檔案內容可編輯修改(注意有關定義、索引定義及文件識別等項目)和儲存(詳參3.3.2)。 檔案編輯完畢後，系統會進行一連串的動作，檢視在SGML2GTI中所用到的標記符號，在DTD中是否都有正確地設定，之後會開啟MEF-GTI和INDEX GTI二個組態設定檔，索引程式會自動確認其內容，如果沒有錯誤發生系統會詢問所產生的檔案是否要儲存，若回答為”是”，便會產生組態設定檔畫面中可供修改編輯的各檔案，這些檔案包括書目資料檔及MEF-GTI、INDEX-GTI二個組態檔，詳參4.1。 最後會產生報表檔並存在“*cfg*”目錄下，有關報表檔在下章**資料準備(Data Preparation)**中會有完整的說明。
Abstract file Modification	說明光碟片中的內容，可以是產品概略式的整體性說明，也可以針對每期的內容作詳細說明，本範例中，檔案是必須依照每期內容在**資料準備(Data Preparation)**的功能中更新。
Copyright file Modification	版權內容說明。
Bibliographic file Modification	Mimosa軟體中使用的檢索功能說明，由於這個檔案內容是配合*SGML2GTI modification*而被啟動，為求其一致性，建議最好不要修改檔案內容，若系統顯示發生錯誤時，可再針對*SGML2GTI modification*做更正。
ISO9660 Configuration file Modification	檔案**iso9660.cfg**可從目錄中“*cfg*”選定，檔案內容可供編輯修改，編輯後的內容會經系統確認過儲存在“*cfg*”目錄中。 註：Note: during Data Preparation additional updates follow (see below).

German French English stop word files modifications	在資料檢索時，有些字因出現頻頻過多而不適用於檢索，一般通稱為停用字(stop word)，這些停用字紀錄在檔案中，可隨時更新。這個檔案在產生索引資料的時候會使用到，檔名在*SGML2GTI modifications*時會設定，除了德法英三種語文外，(專利)文件中如有其他語文之停用字亦可納入。
Mef_GTI Index GTI Configuration file Modification	這二個檔案是由於對*SGML2GTI modification*內容做修改而產生的，修改方法同一般的檔案編輯，但建議勿隨意修改此檔案內容，如需更新建議要由對GTI索引具相當了解的人員來修改。

上述檔案之使用於***資料準備(Data Preparation)***章節中再行討論。

第四章　資料準備

本項有關**資料準備(data preparation)**部分的軟體功能是針對每期專利資料，建立MIMOSA系統下一些特殊的資料集合。

註：由於畫面中各選項之設定要視作業環境情況不同而作不同的設定，且各工作項目確認後皆由系統自行執行相關之程式，因此保留原文，中譯部分僅針對各標題項目簡要說明。

4.1 資料準備工作內容

在前一章節中討論到如何對資料集合建立環境設定，以及利用正確的***.GCD**檔案啟動*GENCD*作業(如範例中之epfd.gcd)。主功能選項畫面如3.4.4所示，請選擇*Setting environment variable(s)(設定系統變數)*選項，大部分的情況下，只須要變更 **ncd=** 欄位的內容為新的期數即可，請詳參3.4.6。

在各選項設定完成且存檔後，再選擇資料準備鏈結*Data preparation chain*的選項，此選項之前有一個未字母A，表示正設定運作中的狀態下，其下系統會展開一連串的子工作項目(3.4.8及3.4.9)如上。

上列各項內容會視環境設定不同而異，例：修改**MMMTDATA=YES, ST33DATA=YES 及 ST30DATA=YES**，將會顯示載入ST30和ST33的檔案等不同的系統訊息，各工作選項可整體(數字鍵0)也可單項設定(數字鍵9)為運作中(項目前顯示A)或非運作中的狀態。工作項目確認執行後便會依照設定條件開始執行動作(詳參3.4.9)，而系統也會視情況需要自行產生工作目錄，目錄名稱系統會自行參考環境變數的設定值。

4.1.1 資料的讀入－Input MIXED MODE data loading

每期要製作在光碟片上的新資料，可由不同的媒體(例如磁帶)讀入到硬碟，儲存在檔案中。

This action reads the input tape(s) from the tape medium specified as “drive=“. This can be any tape medium connected to the system: 1/2”, IBM cartridge, 4mm DAT, 8mm EXABYTE, etc. The tape is read and the contents (when it is Mixed Mode) is stored as a file on hard disk. The file name is tape0.mxm. When more than one tape is read then the digit is incremented (tape1.mx*m,* etc.). But if the file is stored over more than tape, i.e. when it is a multivolume tape file, then the records are appended to one file. If the input file comes from a different medium or is not stored in IBM format, then this step should not be used but the file should be copied in different ways to the _in directory, e.g. with a **tar** command when the input is on tape, or downloaded from a network or when on CD with a copy command.

4.1.2 混合模式之資料轉碼－MIXED MODE datas transcoding

原本所有儲存為混合模式的資料檔案(*.mxm)，在此會依資料型態自動轉成兩種不同的編碼，SGML的文字碼檔案(*st30sgml.sg*m)，和TIFF圖形碼檔案(*st33tiff.ti*f)，同時會自動建立*st30sgml.ptr*及 *st33tiff.pt*兩個相對應的指標檔。

Here are all the mixed mode files with the extension: .mxm from the _in directory processed and split up in one SGML coded text file (*st30sgml.sg*m) and one TIFF coded image file (*st33tiff.ti*f), at the same time the pointer files *st30sgml.ptr* and *st33tiff.ptr* are created. The files are stored in he directories: _sg directory for the SGML and it's pointer file and _tf directory for the TIFF and its pointer file.

By means of the *st33tiff.ptr* file it will be checked for duplicate images. If found the pointer(s) to these image in the file *st33tiff.tif* are stored in the file *dupptiff.tif* and reported (see next paragraph). The results of this action will be reported in the files *mxmtrans.sta* and *mxmtrans.spy*, and a cumulated report will appear in the file *mimosa.sta*.

The usage of the pointer files will be discussed below in *Data merging*.

4.1.3 EP專利文件之定義－Prepare EP Publ for parse

此部分為EP專利文件中，在SGML碼中的特用定義格式。

This is a special set of modifications on the SGML coded input for EP documents. It is left in this manual to show that it is possible to add tasks to the GENCD menu. In this example the file *st30sgml.sgm* is changed in a temporary file and then restored again as *st30sgml.sg*m. No statistics or reports are produced.

4.1.4 DTD語法檢查－Parsing with DTD

使用已確認的DTD內容，逐一檢查每一筆資料內容是否正確，系統會自行產生執行結果報表檔*parse.sgm*，資料如有錯誤則會紀錄於*parse.err*報表檔中，執行後的相關資訊及統

計數據則分別儲存於*parse.spy*及*parse.sta*二個檔案中。

The file *st30sgml.sgm* is validated (parsed) against the DTD specified in the environment variables (dtd) and found in (cfg). As a result a canonical file *parse.sgm* is created. The reason for creation of a canonical file is that the Mimosa formatter can take special actions based on the existence of end tags. Erroneous documents are stored in the file *parse.err*, reports and statistics appear in *parse.spy* and *parse.sta* respectively. *Mimosa.sta* will be updated.

4.1.5 指標及資料查核－Pointer and Checksum generation

此項步驟執行後會產生一個*parse.ptr*的指標檔，以便指出*parse.sgm*中的文件位置，這個指標檔會查核是否有重覆的資料，查核完畢並會累計筆數將結果存入*parse.chk*檔中，提供後續*ISO9660 generation*步驟使用。

The following actions will be carried out:

- A *parse.ptr* file will be created with id's and locations of the documents in the *parse.sgm* file from the previous task.
- A check will be done in this pointer file on duplicate documents. If found the pointers to these SGML documents are stored in the file *duppsgml.ptr*.
- A checksum will be calculated for every document in the file. These checksums will be stored in the file *parse.chk*. This file will be used in *ISO9660 generation.*

The file *mimosa.sta* will be updated. No other report files are produced.

4.1.6 鄰字檢索處理程序－Proximity processing

本項工作選項是用於產生使用鄰字檢索時，所需要的相關索引資訊。

The task *Proximity processing* will only be active when one or more indexes are specified with ***proximity=yes*** in the *sgml2gti.cfg* configuration file.

For the calculation of the distance between words in sentences and paragraphs in a document new temporary SGML tags are inserted in the file. These tags indicate the start of a word, sentence, paragraph and (sub) document. The following actions are performed:

➲ ➲ The average distance of words in a document, paragraph and sentence in the sgml file is calculated. These averages are used for setting of the increments when a new document, paragraph or sentence is found.

➲ ➲ Based on these averages an algorithm is calculated for the number ranges for documents, paragraphs, sentences and words. E.g. increments for document count = 4000, paragraph= 400, sentence = 40 and words = 1.

➲ ➲ Based on the above algorithm the tags for words, sentences, paragraphs and documents are numbered. This numbering will be used during indexing to establish the possibility for proximity searching.

➲ ➲ The file with the proximity tags will be stored as *number.sgm*. Report files are: *proximit.spy* and *proximit.st*a. Other files created are: *pegtbl.peg* and *pegtbl.ptr*I, used in *Data merging*.

4.1.7 SGML轉換為GTI－SGML to GTI transcoding

無論是否有執行前項之**鄰字檢索處理程序**，系統會自行將*number.sgm*或*parse.sgm*輸入檔案轉換為GTI格式，輸出檔名稱為*sgml2gti.gt*i，統計數據及錯誤報告檔名稱則分別為sgml2gti.spy, sgml2gti.sta及 sgml2gti.err。

Either the file *number.sgm* (when the “proximity” option was set) or the file *parse.sgm* (no proximity) is used as input. The documents in the input file are converted to GTI input format for indexing (see below). The output file is *sgml2gti.gt*i. Statistics and error files are: sgml2gti.spy, sgml2gti.sta and sgml2gti.err.

4.1.8 資料合併－Data merging

在此工作項目執行後，會產生第一個終結輸出檔案名為*image*，其中包括文件的混合模式格式及檢索昤會用到的相關索引及訊息。

With this task the first final output file is created. The file is named *image* and it contains the documents in mixed mode format and the tables containing the proximity pointers that make it possible to highlight words when “proximity searching” is done in Mimosa. The following actions are performed:

- By means of the file *parse.ptr* all SGML documents are read from *parse.sg*m.
- Based on the presence of image tags (e.g. EMI or RTI) in the SGML documents, the TIFF images are read from the file *st33tiff.tif* the images are found by means of the file *st33tiff.pt*r.
- The sgml document and the images are written into the *image* file. This layout of the file is conform the ISO9660 standard for CD-ROMs. This means that the SGML document is divided in sectors of 2048bytes and the last part of the document, if less than 2048 is filled with the *pegs table* (see below) and/or padded with NULL characters to a length of 2048. Each sub sequent image starts on a sector and is also divided over sectors of 2048 bytes, where necessary filled with the *pegs table* and/or padded with NULL characters.
See also the schema below.
- The third file to be stored in the *image* file is the so-called *pegs table.* The pegs table contains the physical location of each *peg* in the SGML documents. These pegs are built during *proximity processing* and used for highlighting of words that are searched for in Mimos*a.*
The pegs table stored after the last part of the document if space is available in the sector, if no space available then it is stored in the first free space in a sector containing the end of an image. See also the schema below. For a detailed description see (21).

Layout of the image file

During this process the file *merge.gti* is created. This file contains address pointers to the documents in the image file. These pointers are stored during indexing (see below). By means of these pointers it is possible to read the documents from the CD-ROM directly avoiding access via a long directory structure (see below: *Exchange index).*

When in the environment **image=open** is set, then the file change.idx is built up containing the complete path to the documents. This file will be discussed in: *Exchange index.*

Finally the files *merge.sta* and *merge.spy* are created, containing statistics and (error) reports about missing images.

4.1.9 欄位提示資訊之對應位址－Pointers injecting into notices

產生將來資料製作完畢後，在提供檢索瀏覽時註記資料位址之相關資訊。

The input file *sgml2gti.gti* containing the bibliographic data in GTI input format and *merge.gti* containing the address pointers to the documents in the image file are merged. The new file is *inject.gti* this will be the input for indexing.

During the merge it is checked if for every document in the file *sgml2gti.gti* an entry exists in the file *merge.gti* and visa versa. Missing entries are reported in the files *merge.rej* or *sgml2gti.re*j.

4.1.10 GTI索引－GTI preparation for indexing (MEF-GTI).

產生GTI格式之索引資料。

This is the first step of the indexing where the GTI fields are prepared for indexing. The input is the file *inject.gti* from the previous step. The output is written in the file *mimosa.gt*i.

4.1.11 文字索引排序－Text indexing (Index-GTI)

產生*SGML2GT*檔案中所需要的相關索引資料，並儲存於*index.cfg*檔中。

This process creates the indexes and notices defined in *SGML2GTI* and stored in the file *index.cf*g. The indexes are created and sorted, the input is *mimosa.gt*i. The indexes, notices and the pointers to the documents in the *image* file are stored in the file *mimosa.ro*m. The file mimosa.rom is in GTI format.

Statistics can be found in *indexgti.sta* and a comprehensive report is stored in the file *com*o, all in the *_sg* directory.

4.1.12 產生ISO9660檔－ISO9660generation

此工作項甩執行完畢後，會在硬碟中產生“iso image”，即表示*iso9660.cfg*中所有指定的檔案都已匯入，並儲存在“iso image”中。

In this step the “iso image” of the CD-ROM is created on hard disk. This means that all files specified in the configuration file *iso9660.cfg* are selected and stored in the “iso image”. The

actions are as follows:

➲ ➲ The file *iso9660.cfg* is and the existence of the files to be copied to the “iso image” is checked.

➲ ➲ The file change.idx from *data merging* is used to create an exchange index (when the environment variable **image=ope**n).

➲ ➲ The primary volume directory (ISO9660 standard) is created including standard files like “abstract” and “copyright”

➲ ➲ By means of checksums created during the previous steps the file checked on completeness and correctness.

4.1.13 CD-ROM tape generation

在磁帶上產生可用於製作光碟片的資料。

The “iso image” from the previous step can be used to create a master tape that can be used for production of many CD-ROMs. Because of the size of the “iso image” (up to 500 or 650 MB) this is usually a 4mm DAT tape or an 8mm Exabyte tape.

4.1.14 製作光碟母片－Single CD-R creation

視作業環境下提供之軟硬體設備，燒錄光碟片。

Depending on the available hardware and software it is possible to “burn” a single CD_R(ecordable) directly from the “iso image” on hard disk. For example when one makes use of the GEAR software, the following actions are performed:

➲ ➲ A blank CD-R is inserted in the CD recorder

➲ ➲ The GEAR software is started by entering **msgen** from the directory where the software is stored.

➲ ➲ The software detects the blank CD-R.

➲ ➲ With the following command: write /data/out/97001rom/iso x 1 the “iso image” with filename *iso* is written on the CD-R. Some settings can be modified before the actual write starts: the speed (2 times), the option finishing, ...

4.1.15 報表檔及統計資訊－Report files and statistics

針對處理的過程中，系統會產生一些報表檔或統計資訊，供作業人員參考。

The various reports and statistics can be consulted from the GENCD menu but in practice this shall be done with the file manager and editor from e.g. HPVue.

4.2 資料轉換索引－Exchange index

在環境設定中，如有設定**image=open**，則系統便會在光碟中產生一個索引轉換對照檔，用於光碟中不用透過MIMOSA系統即可存取資料之SGML 文字碼及TIFF影像。

When in the environment **image=open** is set, then an exchange index file is created and stored on the CD-ROM. This exchange index is based on a directory structure that has been developed in such a way that it conforms to the ISO9660 standard that itself is based on the DOS directory structure.

The exchange index makes it possible to access documents: SGML coded text and TIFF images on the CD-ROM without consulting Mimosa. Although many directory structures are possible, it is for Mimosa based on the directory structure being in use for ESPACE and described in the WIPO standard ST.40. A layout of the directory structure follows below.

Exchange index (example)

Explanation:

➲ ➲ The area A contains the document identification, in this example: EPA1 03635866 (Office, Kind and Publication number)

➲ ➲ The area B contains the image or text identification: 00000001 (the first image), followed by 00000002 (the second image), followed by 000000003 (the third image), followed by the text file in SGML: SGMLSGML. The image identification is taken from the *ID* attribute from the EMI tag in the SGML document.

➲ ➲ The area C contains the *existence* of .....

➲ ➲ The area D contains the address (offset) of the images and texts in the *IMAGE* file. The address is calculated as a sector number, starting with zero. An explanation follows in the explanation of area E.

➲ ➲ The area E contains the length of the image or text. So in the example above: the first image starts at address 0 and has a length of 301480 bytes. That means that it occupies 14 sectors of 2048 bytes (last block padded with zero's, see "layout image file"). The second image contains in area D 15 (the next sector) and has a length of 2014 bytes. The SGML text starts in sector 72 (offset 2048 * 72= 147456) and the length is 120902.

➲ ➲ The are F contains the directory path and file name of the image or text.

4.3 錯誤偵除—Error handling

在資料製作過程中難免會有一些錯誤發生，如資料重覆或缺少對應的圖檔等，針對系統偵察到的錯誤，會自動產生一個報表檔，作業人員可以根據這個檔案將錯誤資料修正後，再將整個步驟重新執行一次。

During the Data Preparation process many checks on validity and presence of data are carried out. For instance: the presence of duplicate documents or images, the absence of images, the validity of the SGML set, the consistency between the data in the *image* file and the *indexes/notices*, etc.

Errors found during Data Preparation will often lead to invalid databases or pointers from the indexes to the SGML data and the images. It is therefore advised that when the *mimosa*.sta file reports errors, that the step where these errors occur is checked with the *.spy* and the *.err* files and then where possible to correct these errors and re-run the whole process. When errors are found it is most often due to erroneous input data, so that corrections must be made at the source where the input is created.

4.4 計算硬碟容量－Storage capacity on hard disk

在資料處理過程中必須建立索引，且會產生一些暫時性檔案，因此在硬碟中應保留出足夠的作業空間，一般而言，對於一個500MB大小的資料集合，至少需要4倍以上，也就是2GB以上的作業空間。

As can be seen from the above, many interim files are created during the process: the SGML file is also available as a canonical file, an input file for indexing is build, the *image* file and the *mimosa.rom* file and the *iso* file are contained in the directory rom and many report and statistical file are kept. As a rule of thumb one can say that for the creation of a 500 MB CD-ROM, at least 4 times on free storage on hard disk is needed (2 GB).

However, it is possible to set an option in the GENCD environment that indicates that interim

files may be deleted as soon as these are no longer needed. This option: *DELETE=YES* can be set when a collection runs for some time without causing errors. As long as the process is not stable (e.g. input errors in the SGML coded data or images) it is advised to keep the interim files for error checking purposes. When a complete process has been accomplished successfully, i.e. the CD has been made and if needed, back-ups have been made from the input files (in directory _in) and the output (in directory rom) then all the sub directories of this production (see above) can be deleted. For a detailed description of how to calculate the required disk space, see (15).

4.5 計算光碟容量－Storage capacity on CD

在資料燒入光碟之前，一定要先計算資料容量，儲存在硬碟中的資料一片光碟是否放得下，是否可以剔除掉不需要的檔案，或是建立較精簡索引以節省空間，或是存放在兩片光碟中。

Before a CD (ROM) is created it must be checked if the file size of the ISO file (the image of the CD on hard disk) fits on the CD. At present 2 types of CD can be distinguished: 60 minutes (540 MB disc space) and 74 minutes (650 MB disc space). If the ISO file doesn't fit on either of these 2 types, then it must be investigated:

- if the input can be reduced or split over 2 CD's,
- or less or more simplified indexes can be built up,
- or the size of the miscellaneous extra files on CD can be reduced.

An estimate of the expected file size on the CD can be calculated with the following rules, see(22).

This problem can cause serious delays because this must often be discussed with the designers and/or the input providers. Therefore it is strongly advised to build check routines in the

process that creates the input. For rules about the calculation of file sizes see (15).

4.6 考量其他新媒體—Other (new) output media

目前是使用光碟片發行資料, 不久後也許會用更進步的媒體, 如DVD或可重覆寫入的光碟。

At present the main output medium is the CD-ROM. But new media i.e. the DVD (the Digital Versatile Disk) and re-writeable CD's will be available for commercial exploitation soon. Especially the DVD can be of interest for Mimosa applications because of its storage capacity. This capacity is .... However, the description of the standard the ..... book has not been finished yet.

Reference to: ...

Re-writeable CD's .....

第五章　建立資料索引

(註：本章之內容作者尚未完成，故以原文提供)

5. Master indexes (this chapter must be completed)

Mimosa has a possibility to create cumulated indexes or master indexes. A cumulated index contains the index data of previous issues of a collections in a time frame, e.g. a year. This cumulated index can be used to search not only in the latest issue but also in the previous issues of a collection. The index contains therefore also the ID of the CD-ROM where the document can be found; Mimosa will search for the requested CD (e.g. in a jukebox) or ask to mount the requested CD.

A master index is usually stored on a separate CD and contains index data of a series of CD's of one or more collections (assuming that these have the same index types). The cumulated indexes and master indexes are created by reading of the *mimosa.rom* files from the CD's if necessary in combination with a previously created cumulated index file. The procedure is as follows (for details see 23):

Master index creation

第6章 線上工具On line tools

6.1 簡介

在MIMOSA系統下是以混合式資料格式儲存資料，也就是說文字採SGML格式，圖形採TIFF檔案格式，相較於一般的圖形檔，好處在於大幅節省了儲存空間，但麻煩的是混合式資料在輸出(螢幕顯示或列印)時，必須作格式定義，換句話說，由於圖形和文字編碼各自獨立，因此輸出時，從簡單到複雜，可以有各式各樣不同的呈現方式，不同標的的資料集合，可以選擇不同的輸出格式，例如：美國專利文件和歐洲專利局專利文件就不

相同；系統可以讓使用者指定的順序，顯示不同的文件

6.2 資料輸出的格式

在MIMOSA混合資料模式系統下是使用一個叫做**格式設定(formatter/viewer)**的軟體工具來製作輸出格式，這個程式分為兩個部分：**formatter(格式定義)**是用來製作輸出格式；**viewer(格式顯示)**是依照設定的格式，在資料顯示式列印時啟動相關的硬體設備，下列將有更詳細的說明。

不同的資料或文件，可以設定不同的輸出格式，因此系統必須能夠分辨那種資料對應那種輸出格式，那些文件屬於那個特定的資料集合，這些都設定於**collect.ini**檔案中，使用者也可以不用指定格式，而依需求採用自訂格式，MIMOSA系統也提供了自訂格式的設定工具

6.3 格式設定工具

這個格式設定(formatter/viewer)工具分為兩個部分：

➲ ➲ 格式定義 (formatter)

採用輸入時用的SGML文字格式(非圖檔格式)，依使用者選定的輸出形式建立一個格式頁(*DIP —device independent page)*，在格式頁上記錄著文字欄位的顯示位置，並預留出對應圖形顯示的空間，請參考下列結構圖示：

6.3.1 格式定義

格式定義工具共包括三個程式模組：

➲ ➲ 一般模組

採用輸入時用的SGML文字格式，並利用組態設定檔儲存格式頁(DIP- Device Independent Page)內容，SGML特定的標記符號會啟動設定內容，內容組態設定會告訴格式定義工具每個標記符號該執行什麼動作，動作定義可設定於檔案開頭、中間或是結尾的位置(請參考後續詳細說明)。格式定義採用如同程式之區塊式文字描述，區塊可使用巢狀結構，並用不同的字體以識區別。DIP中程式區塊的位置也會在組態設定檔中加以設定，文存中遇到圖形檔的標記符號時，系統會自動留出一塊固定的空白區域，供圖形嵌入，預留區域的大小，可以透過長度及寬度的數值來設定。

當SGML文字遇到表格或數學式的標記符號時，則會自動轉到下列相對應的程式模組。

➲ ➲ 表格模組

當系統遇到表格的起始標記符號時，便會上生一塊區間，準備置入表格，本項的關鍵在於DIP中指定表格正確起迄位置的格式定義，有關組態檔中對格式處理程序的控制，於後另有說明。

➲ ➲ 數學式模組

當系統遇到數學式的起始標記符號時，便會準備出一塊空間，準備插入數學式，本項關鍵在於DIP中指定數學式正確起迄位置的格式定義，有關組態檔中對格式處理程序的控制，於後另有說明。

6.3.2 格式顯示工具(viewer)

系統會依照使用者設定的格式，透過硬體設備及各驅動程式轉換為版面的呈現，同時在預留的圖形空間上嵌入相對應的圖形。

6.4 資料輸出格式(Style sheet sets)

註：本章節為介紹資料瀏覽輸出時之報表格式製作，由於各國專利資料及所欲製作的光碟內容不盡相同，各欄位及圖形輸出之版面呈現方式亦有不同，因此針對不同的光碟產品，應設計符合需求的輸出版面，再加以設定及製作，一但製作完成後不宜任意變更，以免造成新舊資料呈現不一致的情形，以下內容以原文提供參考。

The formatter software is making use of style sheet sets. These style sheet sets are selected from the

collect.ini (see below for details) or selected by the user from the preference menu. A style sheet set is a fixed set of configuration files that are together responsible for the display and print of pages from a document. The files forming together the *style file set* is specified in another file, usually with the extension *.cfg*, for example *fdep.cfg* and it will
contain the following lines:

fdep.cfg example

➲ ➲ The $FONT_FILE keyword refers to the file containing the specifications of the (TrueType) fonts to be used. These can be standard or private font libraries.

➲ ➲ The $ENTITY_CHARACTER_FILE keyword refers to the file containing the *character entities* with font specifications for display and print.

➲ ➲ The $TEXT_STYLE_FILE keyword refers to the file containing specifications of the various possibilities of font display and print: font size, graphic rendition.

➲ ➲ The $PARAGRAPHE_STYLE_FILE keyword refers to the file containing specifications about the way text parts are formatted.

➲ ➲ The $LAYOUT_STRUCTURE_FILE keyword refers to the file containing the layout specifications for the first page and following pages (where appropriate).

➲ ➲ The $PROCESSING_FILE keyword refers to the file containing the processing instructions for the document, with the SGML tags as trigger.

➲ ➲ The $MATH_CONFIG_FILE keyword refers to the configuration file containing the

style sheet set for the Mathematical sub formatter.

➲ ➲ The $TABLE_CONFIG_FILE keyword refers to the configuration file containing the style sheet set for the Table sub formatter.

All above files and their mutual relations will be discussed in the paragraphs below by making use of the first page example.

6.4.1 The Processing file

The instructions and specifications in the style sheet sets are fully based on the page layout design as discussed in chapter 3.2.3. One must know:

➲ ➲ where bibliographic and text elements must be placed on a page,

➲ ➲ in which order

➲ ➲ under which conditions and

➲ ➲ how it must be formatted

Note: some text elements may be ignored!

> Attention: The formatting process is triggered by the presence of SGML tags in the document. Because of the possibility to display or print sub documents separately or in any sequence, it is necessary to control this process at sub document level. The only tag that is always processed at the beginning of a document (regardless which sub document comes first) is the document tag. So initial settings at the beginning of a (new) document can be specified best at document level.

The most important file from the style sheet set is the *processing fil*e. This file contains the processing instructions for the SGML tags (except for table and math sub tags, see below for details) and shall refer to the *layout file* (where on the page?) and *paragraph style file* (how formatted?). The discussion below will use the *processing file* as a starting point for explanation of all the configuration files in the style sheet set.

The first tag to be discussed is the <PATDOC> tag. The *processing file* contains the

following:

The PATDOC tag in the processing file

6.4.1.1 Discussion of keywords

Tag /Tag

The first line contains: <Tag name=PATDOC>.

The keyword *Tag name=* is used to identify for which SGML tag in the document actions specified between the keywords: <*Tag name=* and the <*/Tag*>, are processed. Therefore in principle for all tags specified in the DTD for the SGML set of this collection a <*Tag name=* definition should be present, even when no action is required, i.e. the data from this tag must be ignored. In this case the begin <*Tag name=* is directly followed by the <*/Tag*> (no actions specified).

Note: Tags may contain other tags (nesting), if these tags must be ignored as well, special actions are required. An example follows later.

Begin Content End

The specifications of processing actions can be set on three occurrences:

- <Begin> : The actions specified in the following lines are executed as soon as the

tag is found.

➲ ➲ <Content> : The actions specified in the following lines are executed on the contents of the tag.

➲ ➲ <End> : The actions specified in the following lines are executed when the end tag is found.

In this example for the *PATDOC* tag actions are specified only at the beginning.

Do

The *Do* keyword is the starting point for the actions to be executed. This can be:

➲ ➲ *Unconditionally:* <Do> this statement follows directly the <*Begin*>

➲ ➲ *Conditionally:* a set of statements indicating the condition and ending with a <Do> sets the condition (examples will be discussed later).

In this example the <Do> is unconditional.

Define Attribute String First Last

The following lines contain a number of initial definitions and settings of fields that will be used to control the display and print of the document. The definition of the fields is done with the keyword: <Define name=...>, the setting is done in various ways:

➲ ➲ *No specification:* the field is set as a *null* field.

➲ ➲ <Attribute name= ..>, the content of the specified *attribute* from the SGML tag *(PATDOC)* is stored in the field (if present). It is possible to store a part of the content by using the keywords: First= and Last= indicating the start and the end of the string to be stored.

➲ ➲ <String value= "..">, the value between the quotes is stored in the field.

➲ ➲ <Variable name= ..>, the content of the specified *variable* (earlier defined and filled field) is stored in the field. It is possible to store a part of the content by using the keywords: First= and Last= indicating the start and the end of the string to be stored

Comments

For documentation purposes comment lines can be added in two possible ways:

// At the beginning of the line turns the whole line into a comment line

/* … */ All the data in between the /* and */ is considered to be comment. For practical reasons it is advised to place the *begin* and *end* comment tokens at the beginning of a line.

6.4.1.2 The PATDOC tag example

In the example the following actions are executed:

- The document identification is saved in Docnum, Office and Kind
- The publication date is saved in DATE
- The field BIBACT is set to “N” (no). It is set to “Y” when the sub document bibliography is displayed. It is checked when the sub document abstract is displayed. When it is “N” then the abstract was not preceded by bibliography and a skip to a new page must be made. If “Y” then the abstract can displayed directly (following the bibliography).
- The fields Level1, Level2, Item1, etc. are used in (nested) lists. Here these are initialized.

6.4.1.3 The SDOBI set

In the following chapter some tags and actions are discussed from the First Page example as a further demonstration of the possibilities in the style sheet set.

Processing instructions for Heading of First Page

<Tag name=SDOBI>

<Begin>

<Do>

<CreateLayoutObject name=”EPFst”>

</Tag>

When the sub document tag SDOBI is triggered, then a new page must be set, this is done with: *CreateLayoutObject name*=. In this example a reference is made to the page with the name “EPFst” in the Layout Structure file:

See 6.4.2 for more information about the Layout structure file.

<Tag name=B100>

</Tag>

The tag B100 is a group tag and requires no action.

<Tag name=B110>

This tag contains the publication number; this number must be displayed together with the office name and kind. The values for these two fields were already saved from the *PATDOC* tag and are contained in the fields *Office* and *Kin*d. So all ingredients are available to build the header part of the page, which must be displayed as follows:

The actions to create the above are:

➲ ➲ At the *Begin* the Inid code (19), the EPO logo and the office name and the Inid code (11), are formatted.

➲ ➲ At the *Content* level the publication number and the document kind are formatted.

<Define name="Hotlink">

<Append name="Hotlink">

<String value="epo_logo.bmp">

The function *hotlink* is used in Mimosa to create a dynamic link to an image. This results on line in the display of a “little hand” when the cursor is in the area of an image. The image can now be displayed and manipulated in a separate window by double clicking on the image. The settings are made by defining the variable *hotlink* followed by an *append* of the *id* of the image to the *hotlink* variable.

The function *hotlink* is used in Mimosa to create a dynamic link to an image. This results on line in the display of a “little hand” when the cursor is in the area of an image. The image can now be displayed and manipulated in a separate window by double clicking on the image. The settings are made by defining the variable *hotlink* followed by an *append* of the *id* of the image to the *hotlink* variable.

<DisplayText ParagraphStyle=sInid LayoutObject=EPLogo>

<String value="(19)">

Now the Inid code (19) must be displayed in front of the logo. The keyword is *DisplayText* and the attributes are *ParagraphStyle* and *LayoutObjec*t. These attributes set:

➲ ➲ The **style** definition *sInid* is further specified in the $PARAGRAPHE_STYLE_FILE, in

the example the file *fdep_par.tx*t:

Style sheet for Inid codes

An explanation of the style definition will be given in chapter ... **The Paragraph Style File.**

➲ ➲ The **layout** definition *EPLog*o, i.e. where on the page shall the logo be displayed or printed,

is specified in the $LAYOUT_STRUCTURE_FILE,

in the example the file *fdep_lay.txt:*

Layout description for Inid code and Logo

An explanation of the layout definition will be given in chapter .. **The Layout Structure File.**

Results so far

In the above example the Inid code (19) is displayed in 10 points Helvetica with a negative indent of 8 (8mm left of the margin), left aligned in the block EPLogo. This block starts at 25mm from the left and 15mm from the top of the page, the measurements of the block are 28mm width and 18mm height and it contains a borderline on top and at the bottom, conform the example above. The next step is the definition of an empty block and the setting of the identification of the image (the EP logo) that must be filled in the empty block.

Note: the viewer will fill the image in the block.

```
<DisplayImage LayoutObject=EPLogo>
    <Height>
    <String value=170>
    <Width>
    <String value=170>
    <Identifier filename=yes>
    <String value="epo_logo.bmp">
```

The keyword *DisplayImage* contains the attribute *LayoutObject* and it tells in which block the image must be placed. With the keywords *Height* and *Width* an empty block is created. In this case the space is known, so the keywords are filled with the value 170. The keyword *Identifier* tells that the image is contained in a file, the filename will contain the value *epo_logo.bm*p, and the viewer will use this filename for insertion of the image in the empty block.

```
<DisplayText ParagraphStyle=sFixNam LayoutObject=EPOffice>
<String value="Europäisches Patentamt">
<block name="EPLogo"
PosType=Fixed
```

```
x=25 y=15
width=28 height=18
topBorder=YES bottomBorder=YES>
</block>

    <Break>
    <String value="European Patent Office">
    <Break>
    <String value="Office européen des brevets">
    <Break>
    <DisplayText ParagraphStyle=sInid LayoutObject=EPPubNum>
    <String value="(11)">
```

With the above lines the name of the patent office is filled in the block *EPOffice* and with the style *sFixNa*m. The keyword *Break* causes a line break so that the three names appear at a different line.

With the last two lines the block EPPubNum is set and the Inid code (11) is filled in the style *sIni*d. It will be followed with the publication number.

```
<Content>
    <Do>
    <DisplayText ParagraphStyle=s16BR >
    <Variable name=Office>
    <String value=" ">
    <TagValue First=1 Last=1>
    <String value=" ">
    <TagValue First=2 Last=4>
    <String value=" ">
```

<TagValue First=5>

<String value=" ">

<Variable name=Kind>

For the creation and display of the publication number the saved office name from the <PATDOC> tag is used *(Office)* and the kind code *(Kind*), the number itself is selected from the tag <B110>. This is the reason why the actions take place based on the keyword *Conten*t. The publication number uses the style sheet *s16B*R, that is: Helvetica 16 points, bold and right justified:

The publication number is made up with spaces between the office, number and kind, and also the number is split up with spaces. The split is done by *TagValue First=2 Last*=4, i.e. positions 2,3 and 4 are selected from the input and then a space is inserted *(String value=”* “).

Results

The result of these actions is the display of the header of the as shown in figure …

6.4.3 Other processing instructions

In this paragraph it will be discussed how the various instructions in the example are used. The purpose is to give a feeling for the many possibilities that are offered. A full description can be found in (5).

The conditional construction

```
<style styleName=s16BR unit=MM
```

```
textStyle=sHelveticaBold16
lineSpace = 0.4
indentation=0
alignment= right
spaceTop=3 spaceBottom=0 spaceLeft =0
spaceRight=0
firstLineOffset=0>

<If>
<Compare condition=EQ>
<Variable name=Lang>
<String Value="F">
<Compare condition=EQ>
<Variable name=Kind>
<String Value="A3">
<Do>
<Define name=LastItem>
<String value="43">
<DisplayText ParagraphStyle=sInid LayoutObject=PubDat>
<String value="(88)">
<DisplayText ParagraphStyle=sFixTxt>
```

<String value="Date de publication A3: ">

The conditional construction exists from the keywords:

- *If*: the start of the condition. An *If* may contain one or more comparisons.
- *Compare condition*=. This starts a comparison between two values and it indicates the condition:

EQ	Equal
NE	Not Equal
LT	Less Than
GT	Greater Than
LE	Less or Equal than
GE	Greater or Equal than

➲ ➲ Variable name= / String value= / Resource name= / Attribute name= / ListItem name= :

any combination of two fields from this set is allowed.

When the *If* is true then the actions following the *Do* are executed.

An *If* construct may be followed by an *If* an *ElseIf* (another *If* but excluding the previous *If(s)*) or an *Else*) unconditional but excluding the previous *If(s)*).

More about images

```
<DisplayImage ParagraphStyle=sEMICF factor=7.5 viewerstyle=1>
          <Height>
             <Attribute name=HE>
          <Width>
             <Attribute name=WI>
          <Identifier>
             <Attribute name=ID>
```

Apart from what was discussed with the “logo” example, the following can be set:

➲ ➲ *Facto*r: used for reduction of the image. The *factor*=10 gives the original size, in the example the *factor* is set to 7.5 and the result will be that the image size is reduced to 75%.

➲ ➲ *Attribute name=:* the values from the attributes *HE, WI* and *ID* are stored in respectively: the Height, the Width and Identifier.

Tables and formulae

```
<Tag name=TAB table=yes ParagraphStyle=sEMIAD>
```

```
<Tag name=DF mathematical=yes ParagraphStyle=sEMIAD>
```

When a document contains tables or mathematical formulae, then the special formatter for tables or math must be activated. This is done by definition of the start tag (TAB or DF or DFG or F) and addition of the attribute *table=yes* or *mathematical=yes*. All tags in between the start and the end tag are triggered by the table or the mathematical formatter.

6.4.2 The Layout Structure file

The Layout Structure file contains the layout specifications for the first page and the following pages (where appropriate). The specification is done by the definition of blocks that can be filled with text and/or images.

The specifications in the Layout Structure must be made from a page design. The page design is based on a page defined in blocks with a specific character and/or at a specific place on the page. Blocks may contain other (sub) blocks. This is illustrated on the First Page layouts below:

The first page illustrated above can be divided in the following blocks with specific properties:

The blocks in blue colors are main blocks; the blocks in red are sub blocks and are for special purposes within the main blocks, i.e.:

➲ ➲ Store fields in a specific part of the block, this is the case in the main block EPBib1 (to be discussed below).

➲ ➲ To fill columns in a block and balance the contents over the sub blocks, this is the case in the main block EPBib2 (see also below).

➲ ➲ To fill columns in a block by filling the first column completely and the remaining data in the next column(s), i.e. the news paper style. It is not used in the first page example.

6.4.2.1 The layout file

A page layout description starts with the page definition:

```
<page name="EPFst"
      type=OrderedRows
      height=297 width=210
      topMargin=15 bottomMargin=10
      leftMargin=25 rightMargin=20>
```

The keyword <*page* starts a page definition and it contains the following attributes:

➲ ➲ *name*= this name (e.g. EPFst) is used in the processing file (see chapter 6.4.1.3) to activate a (new) page.

➲ ➲ *type*= it indicates how the page will be built up, i.e. *OrderedRow*s, the (main) blocks

can be considered to be horizontal rows.

➲ ➲ *height*= *width*= specify the size of the page (A4)

➲ ➲ topMargin= bottomMargin= leftMargin= rightmargins= these specify the margins on the page.

The EPLogo block

```
<block name="EPLogo"
              PosType=Fixed
               x=25 y=15
              width=28 height=18
              topBorder=YES bottomBorder=YES>
       </block><block name="EPLogo"
              PosType=Fixed
               x=25 y=15
              width=28 height=18
              topBorder=YES bottomBorder=YES>
</block>
```

This block is the definition for the display of the Inid code (19) and the EPO logo and it contains the following attributes:

➲ ➲ *PosType*= it tells how the block must be positioned in relation to the father. The default value is *variabl*e, which means that the block will start where the previous block ends (in a vertical sense). In this case it is *Fixe*d: the block must start at a fixed place.

➲ ➲ *x*= *y*= the start coordinates measured from the upper left corner of the father. In this case it is the page and it starts on position 25 and 15 respectively (taking into account the margins).

➲ ➲ *width*= *height*= this is the size of the block

➲ ➲ *topBorder*= *bottomBorder*= *(leftBorder*= *rightBorder=)* these attributes specify if

one or more of these borders must be set and if so, which type of border lone must be used. In this example *YES* is given for the top and bottom border and both will be set to a single line.

Note: the blocks *EPOffice* and *IEPPubNum* start at the same vertical level *(y=15)* but with different x coordinates. *EPOffice* contains only a top border.

The EPBib1 block

```
<block name="EPBib1"
              type=OrderedColumns
              x=25 width=182
                   topMargin=3 bottomMargin=3>
              <block name="PubDat"
                   PosType=Fixed
                   width=82>
              </block>
              <block name="EPClfix"
                    PosType=Fixed
                    x=82 width=100>
              </block>
</block>
```

The *EPBib1* is an example of a block containing blocks to be used for the placement of data on a specific place. It contains the attribute *type=OrderedColumns* i.e. it contains one or more *columns* and *ordered* as specified *(unordered* is also possible, then the placement of the column depends from the moment it is filled).

The main block contains no *PosType* attribute and shall therefore start where the previous block ends; in this example after the block *EPPatNa*m.

The block *PubDat* and *EPClfix* contain a *PostType=Fixed* and have therefore fixed places within *EPBib*1:

➲ ➲ *Pubdat* starts in the upper left corner in *EPBib1* (no x and y coordinates are given). The width of the block is 82 mm.

➲ ➲ *EPClfix* starts in top of *EPBib1* (no y) but 82 mm from the left (x=82).

In this example no column relation is specified for the sub columns. This means that the columns must be filled independently, i.e. the data is written directly in the block (the name of the sub block is used in the *LayoutObject* attribute of the *DisplayText* keyword in the Processing instructions). *PubDat* shall contain publication numbers and dates, *EPClfix* shall contain classification codes and where appropriate numbers and dates related to the WIPO publication (Inid 86 and 87).

The EPBib2 block

```
<block name="EPBib2"
                type=OrderedColumns
                x=25 width=165
                ColumnsRelation=Balanced
                bottomMargin=2>
                <block name="EPBal1"
                        PosType=Fixed
                        width=82
                        rightMargin=1 rightBorder=YES
                        topMargin=3 bottomMargin=3
                        topBorder=YES bottomBorder=YES>
                </block>
                <block name="EPBal2"
                        PosType=Fixed
                        x=82 width=82
                        leftMargin=1 leftBorder=YES
```

```
                    topMargin=3 bottomMargin=3
                    topBorder=YES bottomBorder=YES>
            </block>
    </block>
```

The *EPBib2* is an example of a block containing blocks that must be filled so that both columns contain the same amount of data. The attribute for this is *ColumnRelation=Balance*d. This means that the data is divided over the two sub blocks, therefore the data must be written to the main block *EPBib2* (used in the *LayoutObject* attribute of the *DisplayText* keyword in the Processing instructions).

The positioning of the two sub blocks *EPBal1* and *EPBal2* within the *EPBib2* block is the same as it is for the sub blocks in *EPBib*1. But the sub blocks have a top and bottom border and a vertical border between the two (a right border for *EPBal1* and a left border for *EPBal*2).

6.4.3 The Paragraph, Text Style and Font files

Texts must be provided with a style for presentation. In Mimosa this is done via three configuration files:

- The *Paragraph style* file.

This file contains the style specifications. The processing file refers to the attribute styleName of a *style* definition. The *style* contains specifications about the *placing* of the text and it refers to a style definition in the

- *Text Style*.

This file contains the graphic renditions for the fonts being used and it refers to the

- *Font* file.

This file contains the specifications of the fonts being used. These are the standard Windows fonts being used and the specific font *Mayenn*e. The Mayenne font file contains special

characters not available in the (standard) Windows fonts.

The example below illustrates this with the definition for the style of *sInid* (the Inid code):

➲ ➲ The instruction in the *processing* file:

```
<DisplayText ParagraphStyle=sInid LayoutObject=EPLogo>
<String value="(19)">
```

➲ ➲ Refers to the style *sInid* in the *Paragraph style* file:

```
<style styleName=sInid unit=MM
       textStyle=sHelveticaNormal10
            lineSpace=0
            indentation=-8
            alignment=left
            spaceTop=3 spaceBottom=0 spaceLeft=0 spaceRight=2
            firstLineOffset=0>
```

➲ ➲ The above specification tells that the text must be *left* aligned with a negative *indent* of 8 (8 mm left of the margin), extra *space on top* of 3 mm and *space right* of 2 mm. It also refers to the *style* sHeleveticaNormal10 in the *Text style* file:

```
<style styleName=sHelveticaNormal10
            fontName=Helvetica fontSize=10
            graphicRendition=normal
            underlined=no overlined=no
            superScript = no underScript=no>
</style>
```

➲ ➲ This tells that the *fontsize* is 10 points, normal *rendition* (can also be bold, italics, etc.).

There are no lines specified above or below the text (possibilities are: single, double, dotted,

etc.). It is also not a subscript or a superscript and the *font* name is Helvetica, referring to the *Font file:*

```
<font userName=helvetica
            typeFace="Arial"
               weight=FW_NORMAL
               outPrecision=OUT_TT_PRECIS
               clipPrecision=CLIP_STROKE_PRECIS
               quality=DRAFT_QUALITY
               pitch=VARIABLE_PITCH
               family=FF_SWISS
               charset=ANSI_CHARSET>
```

The font being used in this example is: *Aria*l.

For more information about style definitions and possibilities see (5).

References

The references in the table below refer to a sub set of documents that can be found on the CD containing the latest version of the Mimosa and the Authoring Data Preparation software. A complete overview of the available can be found in (1).

Reference

number

Document name Directory on CD Description

1 LIST_DOC.DOC \DOC\SOFTWARE Overview of available documents

2 BK1PRT1.DOC \DOC\SOFTWARE\SPECIFIC Mimosa technical Documentation

BOOK1.

Search and Retrieval Software Part 1.

QUICK STARTER GUIDE

3 BK1PRT2.DOC \DOC\SOFTWARE\SPECIFIC Mimosa technical Documentation

BOOK1.

Search and Retrieval Software Part 2.

COMPREHENSIVE USERGUIDE

4 COOKBOOK.DOC \DOC\DATAPREP\SPECIFIC Mixed Mode - Authoring and Retrieval Software

COOKBOOK

5 FMTMUTI.DOC \DOC\SOFWARE\GTIV3 SGML FORMATER CONFIGURATION

USER'S GUIDE

6 GENCD_UX.DOC \DOC\DATAPREP\SPECIFIC Mixed Mode - Authoring Software

Gencd UNIX Version - User’s Guide

7 MEF.DOC \DOC\DATAPREP\GTIV3 Reference Manual MEFGTI Text

Data Preparation Tool GTI V 3.0

Data Preparation

8 INDEX.DOC \DOC\DATAPREP\GTIV3 Reference Manual IndexGTI Data Indexation Tool GTI V3.0 Data Preparation

9 ISO.DOC \DOC\DATAPREP\GTIV3 Reference Manual ISOGTI ISO9660 File Preparation Tool GTI V 3.0 Data Preparation

10 SGML2GTI.DOC \DOC\DATAPREP\SPECIFIC Mixed Mode - Authoring Software SGML2GTI User's Guide

11 UNIXGUID.DOC \DOC\DATAPREP\SPECIFIC Mixed Mode - Authoring Software User's Guide for UNIX Version

12 WIPO ST.32

SGML

Version 3.32 or later

Printed WIPO publication Recommendation for the markup of Patent Documents using SGML

13 Character sets &

entity references

Version 1.0 or later

Printed EPO publication Standard and special characters used in EPO and the way these are coded

14 ST.35 (MMMT

standard)

Version 2.1 or later

Printed publication submitted to WIPO for standard Recommended standard format for data exchange of Mixed Mode published Patent document information

15 STAT_MXM.DOC \DOC\DATAPREP\SPECIFIC Mixed Mode - Authoring Software BenchMarking

16 DOSGUID.DOC \DOC\DATAPREP\SPECIFIC Mixed Mode – Authoring Software User's guide for DOS version

17 TIFF spcefications

18 Rules for contractors

19 SGML Van Herwijnen

20 ISO9660 standard

21 Layout image file

22 Calculation disk size

23 Master index creation