XBRL Global Ledger Framework
~技術規格解析~

周國華
屏東商業技術學院

初版：2007.12.14
本次修訂：2011.4.7

目錄：

主題	頁次		主題	頁次
智慧財產權聲明	3		gl-muc及gl-muc-content	40
閱讀前的準備	4		gl-usk及gl-usk-content	41
XBRL GL 發展歷程	5		gl-taf及gl-taf-content	42
XBRL GL 的特色及目標	6~8		gl-srcd及gl-srcd-content	43
XBRL FR vs. XBRL GL	9		Palette schema文件	44
XBRL GL Framework 的架構	10~12		GL 提供的審計軌跡	45
元素定義與內容模型分開	13~14		XBRL GL Framework 案例文件結構	46~47
使用tuple結構的爭議	15		Context 與 Unit 元素	48
GL為何堅持使用tuple結構？	16		<accountingEntries>	49~50
GL Framework 檔案總管結構	17		<accountingEntries>的內部成員結構	51~64
內容模型的穩定性	18		XBRL GL 案例文件類型	65~73
元素定義套件的結構	19		XBRL GL 分錄類型	74
調色盤套件	20~21		XBRL GL 帳戶訊息	75~80
最單純套件：CASE-C	22~23		明細帳連結訊息	81
CASE-C-B 套件	24~25		Amazing Account	82~83
CASE-C-B-M 套件	26~27		XBRL GL 憑證訊息	84~85
CASE-C-B-M-U 套件	28~29		XBRL GL 過帳狀態訊息	86
CASE-C-B-M-U-T 套件	30~31		XBRL GL 稅務訊息	87~89
CASE-C-B-T 套件	32~33		GL 與帳簿連結訊息	90
CASE-C-T 套件	34~35		GL 與 FR 案例連結訊息	91
最複雜套件：CASE-C-B-M-U-T-S	36~37		GL 案例文件範例	92
gl-cor及gl-cor-content	38		XBRL GL 的應用實例	93
gl-bus及gl-bus-content	39		案例文件結構比較 & 相關網頁連結	94~98

智慧財產權聲明

- 本文件係由周國華老師獨自撰寫，除引用之概念屬於原文作者外，其餘文字及圖形內容之智慧財產權當然屬於周老師獨有。
- 任何機構或個人，在未取得周老師同意前，不得直接以本文件做為學校、研究機構、企業、會計師事務所、政府機關或財團法人機構舉辦教學或進修課程之教材，否則即屬侵權行為。
- 任何機構或個人，在未取得周老師同意前，不得在自行編撰的教材中直接大量引用本文件的內容。若屬單頁內部分內容之引用，亦請註明出處。

閱讀前的準備

在閱讀本文件前，你最好能...

- 對XML Schema、XLink技術規範有完整的認識。可參考「位元文化」編撰的『XML技術實務』(文魁出版)第五章及第六章。
- 瀏覽過XBRL 2.1規格書，包含所有Schema文件。
- 仔細閱讀過周老師編撰的『XBRL 2.1技術規格解析--入門篇』教學文件。
- 瀏覽過GL Framework技術文件。(本教學文件已根據2010-04-12 GL PR做修正。)

XBRL GL 發展歷程

- XBRL GL：Global Ledger (原名稱為General Ledger)。
- GL 1.0：2002/4/10公布，符合XBRL 2.0規格。獲得「approved」認證資格，並達到「final」發展狀態。
- GL 1.1：2003/9/24公布，符合XBRL 2.0a規格。獲得「acknowledged」認證資格，但仍屬草案文件。
  - GL 1.1的技術性文件主要是由日本Hitachi公司的三分一信之(Nobuyuki Sambuichi)執筆，故其內容包含完整的英、日文註解。
- GL 2005：最後版本在 2005/11/7公布(CR)，符合XBRL 2.1規格。GL 2005已被GL Framework取代。
- GL Framework：符合XBRL 2.1規格，已於2007/4/17成為Recommendation文件。
- GL SRCD套件：最新版為2009/5/12公布的2nd PWD文件。
- GL Framework + 日文標籤：最新版本日期為2010/4/12，目前為PR (proposed recommendation)文件。

XBRL GL 的特色

- XBRL國際組織(XII)官方網頁上強調的GL特色：
  - 報告獨立性(Reporting independent)：透過 XBRL GL蒐集的資料可以按照任何格式表達。
  - 系統獨立性(System independent)：任何作業系統平台的軟體皆可透過匯入/匯出功能在現有系統資料與XBRL GL格式資料之間進行轉換。
  - 允許合併(consolidation)：市售中、低階軟體產品對來自不同部門或組織的資料進行合併通常力有未逮，XBRL GL可協助處理此問題。
  - 提供彈性(flexibility)：相較於EDI的侷限性，XBRL GL可提供延伸性、彈性化、跨國性的解決方案。
  - 與XBRL的財報功能互補：XBRL GL可將XBRL財報上的數字與其成因相連結，可滿足審計工作底稿、預算規劃、細目報告之需求。

XBRL GL 的目標 (共2頁)

- XBRL國際組織認為XBRL GL能滿足以下目標：
  - 能向上連結至多套GAAP及各自對應的財報分類標準。
  - 做為第三方軟體的標準格式，以產生日記簿分錄並帶入客戶GL系統內。
  - 做為標準格式，以便將已過帳或未過帳的GL資訊在分支機構與合併系統、預算及預測工具、報告工具之間移動。
  - 做為標準格式，以便在企業進行系統轉換時，將總帳及明細帳檔案及餘額從舊系統移轉至新系統，或上傳至網路ASP系統。
  - 做為標準格式，以便將資訊從客戶系統移轉至會計師查核系統。
  - 做為標準格式，以便將資訊從某項會計師系統(e.g., 帳務)移至另一項會計師系統(e.g., 稅務)。

- 做為標準格式，以便表達應收/應付內容、存貨餘額及其他要和銀行分享的資產基礎衡量指標。
- 做為表達工具，以呈現績效衡量報告項目之細節。
- 可針對任何型態強制性審計軌跡之需要做延伸。
- 可針對任何明細帳之需要做延伸。
- 被設計成符合XBRL規範，但亦能輕易地轉換至其他用途。
- 和一般XBRL案例文件的一項重大差異是：GL案例文件並不在意背景元素(context)所標示的內容，換言之，背景元素的內容無所謂對錯或遺漏的問題。(註：請參閱本教學文件第47頁說明。)

XBRL FR vs. XBRL GL

XBRL FR	XBRL GL
XBRL FR是財務報告端的應用。	XBRL GL是會計帳務端的應用。此外，GL的通用框架也可用來表達會計帳務以外的其他業務明細內容。
案例文件表達的是各科目彙整後的訊息。	案例文件可表達彙整後及彙整前的訊息。
XBRL FR的分類標準通常十分龐大，包含大量的財務報告相關元素定義，以及這些元素彼此間的計算、表達關係。	XBRL GL並未針對個別財務報告元素做定義，而是以一個通用的<account>框架讓GL案例文件編製者自行填入科目名稱、金額、背景等訊息。
背景(context)及單位(unit)元素有實質意義，一份案例文件通常有許多此類元素。	這兩類元素對GL案例無實質意義，因為它們的內容都是GL本身的實質內容而非背景訊息。GL案例內通常各只有一個context及unit元素。

XBRL GL Framework的架構(共3頁)

- GL Framework (2010/4/12 PR版本，含日文標籤)共有七個套件：
  - GEN：通用內容模型套件，此套件內僅有一份schema文件，內含25個通用的項目型態定義(比2009版本多出7個)，以供元素定義套件參照使用。
  - COR、BUS、USK、MUC、TAF：元素定義套件，每份套件內都有元素定義schema及presentation linkbase文件，以及內含英文及日文label linkbase各一份的lang資料夾。Schema文件只做元素定義，元素內容模型則參照GEN或者交由PLT套件內的content schema文件處理。
  - PLT：調色盤套件，有七種不同組合的次套件，每個次套件內包含三份gl-plt schema(可選擇載入英文或日文標籤)及一份或多份與元素定義套件對稱的content schema文件。

- 調色盤套件(PLT)內的次套件均以COR (core)套件為核心套件，再選擇性加上延伸套件。
- 延伸性(add-on)套件：
  - 地域別(jurisdictional)：按各國會計慣例所做的延伸，例如：USK (concepts for the US, UK)套件。
  - 功能別(functional)：按企業營運的個別功能所做的延伸，例如：BUS (advanced business concepts)、MUC (multicurrency)及TAF (tax audit file)等三份套件。此外，XBRL GL工作小組也在研發CMT (cash management)套件。
  - 產業別(industrial)：按不同產業的特殊需求所做的延伸。
  - 其他：視情況需要，在上述類別之外所做的延伸。例如， SRCD套件(目前是草案第二版)，可讓XBRL GL與XBRL FR文件之間的連結更為順暢。

- GL Framework (2009/5/12含SRCD版本) 共有八個套件：
  - GEN：通用內容模型套件，此套件內僅有一份schema文件，內含18個通用的項目型態定義，以供元素定義套件參照使用。
  - COR、BUS、USK、MUC、TAF、SRCD：元素定義套件，總共定義了415個元素。每份套件內都有元素定義schema、label linkbase及presentation linkbase文件。Schema文件只做元素定義，元素內容模型則參照GEN或者交由PLT套件內的content schema文件處理。
  - PLT：調色盤套件，有八種不同組合的次套件，每個次套件內包含一份gl-plt schema及一份或多份與元素定義套件對稱的content schema文件。

元素定義與內容模型分開 (共2頁)

- GL是一個大量使用tuple結構的分類標準：
  - 在GL 1.0及1.1版本時，因為遵循XBRL 2.0規格，tuple元素在定義時並不需要處理內容模型，而是交由definition linkbase來處理。在面對多重套件不同組合下的tuple元素結構調整問題時，可藉由palette schema所連結的definition linkbase文件來做相應的調整。
  - GL Framework遵循XBRL 2.1規格，tuple元素的內容模型必須在schema內定義，但同一tuple元素在多重套件不同組合下可能有不同內容，故必須將元素定義與內容模型分在兩份schema內處理，元素定義schema維持不變，內容模型schema則隨著不同組合而改變內容。
  - 有關tuple元素的內容模型隨著不同套件組合而更動，可參考GLTFTA 1.0或1.1規格書的example 5、6、7。

- 因為元素定義與內容模型分屬兩份schema文件，個別元素定義schema文件並不是一份合法文件，它必須由對應的內容模型schema文件以<include>語法載入，因此這兩份文件必須有相同的目標名稱空間(targetNamespace)。
- 元素定義schema文件(檔案名：gl-xxx-2010-04-12.xsd)置於元素定義套件內，內容模型schema文件(檔案名：gl-xxx-content-2010-04-12.xsd)則置於PLT套件的各個次套件內。
  - 上述xxx代表三個小寫字母的套件縮寫，即cor、bus、usk、muc、taf。
  - 在GL Framework 2009/5/12含SRCD版本中，上述檔案的對應日期則為2006-10-25，SRCD對應日期為2009-05-12且套件縮寫為srcd。

使用**Tuple**結構的爭議

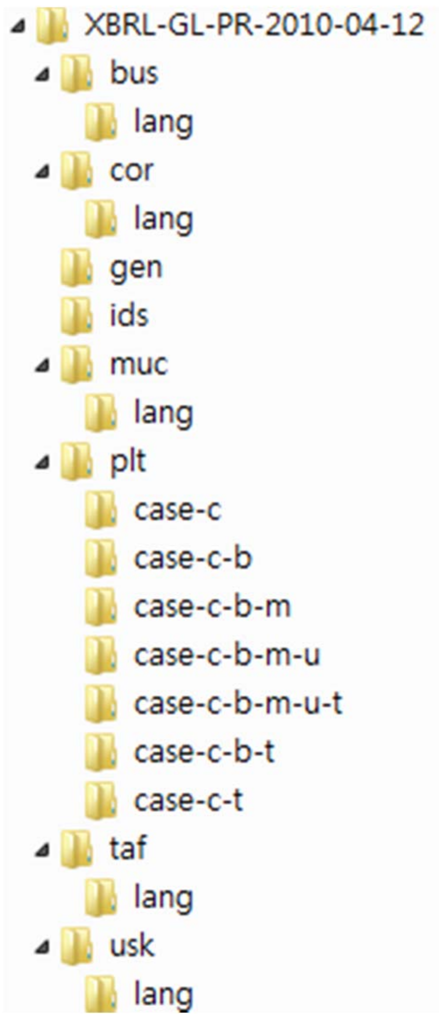
- 在XBRL國際組織公布Dimension規格書後，各國新制訂或修訂的財報分類標準大多數已放棄使用tuple結構，改用dimension來處理原本的內容。
- 目前XBRL主流觀點認為tuple結構有以下缺點：
  - 不容易做延伸：這可從GL Framework必須把tuple的元素定義與內容模型分別放在不同schema內才能進行延伸，就可看出。雖然XBRL 2.1規格書特別制訂了similar-tuples arc來處理tuple的延伸問題，但幾乎沒人使用過。
  - 把元資料(metadata)與資料(data)搞混：tuple其實可被視為是<u>對分類標準做延伸</u>的一種<u>捷徑</u>，在本質上接近元資料，卻被當成資料而呈現在案例文件中。
  - Dimension技術可處理tuple所處理的大部分內容，對市場而言導入兩套功能類似的技術只會造成困擾。
  - 因為導入dimension的案例多，軟體支援性較佳；反之，許多開發團隊認為使用tuple結構會造成軟體開發的極大困擾。

GL為何堅持使用Tuple結構？

- XII的GL工作小組認為GL應繼續使用tuple結構，原因是：
  - Dimension也會混淆metadata和data，只是方向相反(應呈現在案例文件中的資料被定義在分類標準內)。
  - Dimension會把案例文件內大量的內容(content)放入背景(context)元素中，這和一般案例文件是以內容來參照context的慣例不符。
  - Dimension不易處理高度階層化的資料，而這卻是tuple的強項。
  - Dimension把資料放入context元素的segment或scenario子元素內，在本質上都屬於彙整性資料。GL則使用tuple來呈現彙整前的細節，這些細節遠遠超過context元素所能呈現的，所以在GL中context元素基本上不具實質功能。
  - 會計系統的資料模型主要在蒐集彙整前的細節，使用tuple結構可確保細節的完整性。如果使用dimension結構來蒐集資料，會被迫放棄許多有價值的內容。
  - 為了專注於dimension的軟體開發而犧牲tuple及其潛在效益，不具說服力。

GL Framework 檔案總管結構

- 右圖各資料夾內容如下：
  - bus、cor、muc、taf、usk等五個資料夾為元素定義套件，每個套件內含一份元素定義schema、一份presentation linkbase及一個內含兩份label linkbase的lang資料夾。
  - gen為通用內容模型套件，內含一份schema文件。
  - ids內含10份範例案例文件。
  - plt內目前有7種組合(2009+srcd版為8種)，每個組合內均包含三份plt schema文件及一份或多份content schema文件。其中，plt schema為案例文件的DTS起點文件。



內容模型的穩定性

- 為維持內容模型的穩定性，GLTFTA 文件(2009及以前為1.0版，2010起為1.1版)規定以下控制方式：
  - GEN套件的schema文件所定義的內容模型，不准更動。
  - 如果元素定義schema文件內有內容模型的定義，這些內容模型也不准更動。
    - GL Framework本身的五份元素定義schema文件內，僅BUS套件的schema內有內容模型定義(未納入2010版本的SRCD套件的schema內亦有內容模型定義)。
    - 各層級使用者在制訂GL延伸性分類標準時，亦可在新增套件的元素定義schema文件內定義不可更動的內容模型。
  - 內容模型schema文件內的所有內容模型，都可視需要更動內容。

元素定義套件的結構

- GL Framework共有COR、BUS、MUC、USK及TAF 等五份元素定義套件(SRCD尚未納入2010-04-12版本內)。
- 每一份套件，都包含元素定義schema及presentation linkbase文件，以及內含兩份label linkbase的lang資料夾 。
  - MUC套件另有一份definition linkbase文件(參考本文件第87頁說明)。
  - 雖然元素定義schema並不含內容模型(BUS、SRCD除外)，但不影響元素的標籤及表達設定，因標籤及表達設定只與元素名稱有關，而與內容模型無關。
  - 在2009及以前版本中，元素定義schema負責連結presentation及label linkbase文件。在2010-04-12版本中，元素定義schema只連結presentation linkbase文件，至於label linkbase則交由plt套件內的gl-plt schema負責連結。
- 財務報告各會計項目間的計算關係，在GL中並不適用，故GL沒有calculation linkbase。
- GL中所表達的基本會計概念，除了源自Pacioli的雙式簿記外，並沒有權威性的規範性文件，故GL亦無reference linkbase。

調色盤套件 (共2頁)

- 調色盤(PLT)套件內共有七種不同組合的次套件：
  - case-c：內含gl-cor-content及gl-plt-all(同時連結英日文標籤)、gl-plt-en(連結英文標籤)、gl-plt-jp(連結日文標籤)等共四份schema文件。
  - case-c-b：內含gl-bus-content 、gl-cor-content及三份gl-plt schema文件。
  - case-c-b-m：內含gl-muc-content、gl-bus-content、gl-cor-content及三份gl-plt schema文件。
  - case-c-b-m-u：內含gl-usk-content 、gl-muc-content、gl-bus-content、gl-cor-content及三份gl-plt schema文件。
  - case-c-b-m-u-t：內含gl-taf-content、gl-usk-content、gl-muc-content、gl-bus-content、gl-cor-content及三份gl-plt schema文件。

- case-c-b-t：內含gl-taf-content、gl-bus-content、gl-cor-content及三份gl-plt schema文件。
- case-c-t：內含gl-taf-content、gl-cor-content及三份gl-plt schema文件。

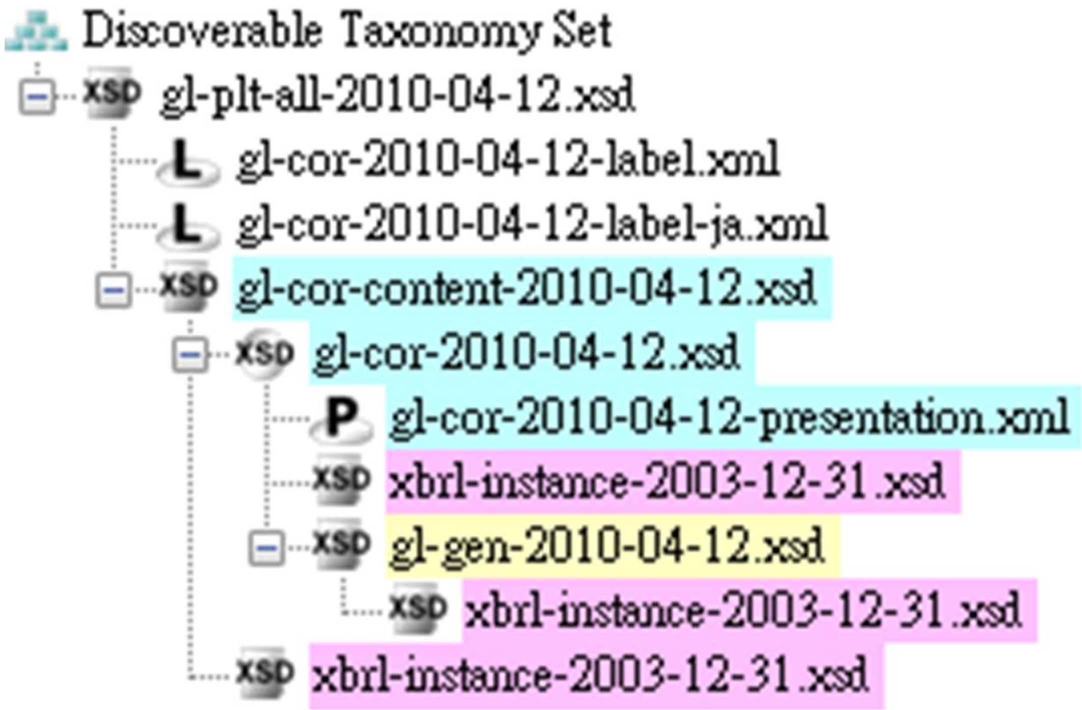
＊在2009+SRCD版本中另有case-c-b-m-u-t-s套件：內含gl-taf-content、gl-usk-content、gl-muc-content、gl-bus-content、gl-srcd-content、gl-cor-content及一份gl-plt schema文件。

- gl-cor-content及gl-bus-content這兩份schema文件在各套件內的名稱雖相同，但內容模型隨套件而異，使用時切忌混淆！！

最單純套件：CASE-C

- 本套件包含四份schema文件： gl-cor-content及三份gl-plt。
- 各文件內容：
  - gl-plt schema文件未做任何元素或型態定義，GL案例文件以本文件做為DTS找尋程序的起點文件。本文件載入gl-cor-content schema文件，並連結至選定的gl-cor英文或日文label linkbase文件。
  - gl-cor-content schema文件定義86種(舊版為93種)內容型態。本文件載入COR套件的gl-cor schema文件。
    - gl-cor schema文件定義119個元素。本文件載入GEN套件的gl-gen schema文件。
    - gl-gen schema文件定義25種通用項目型態。

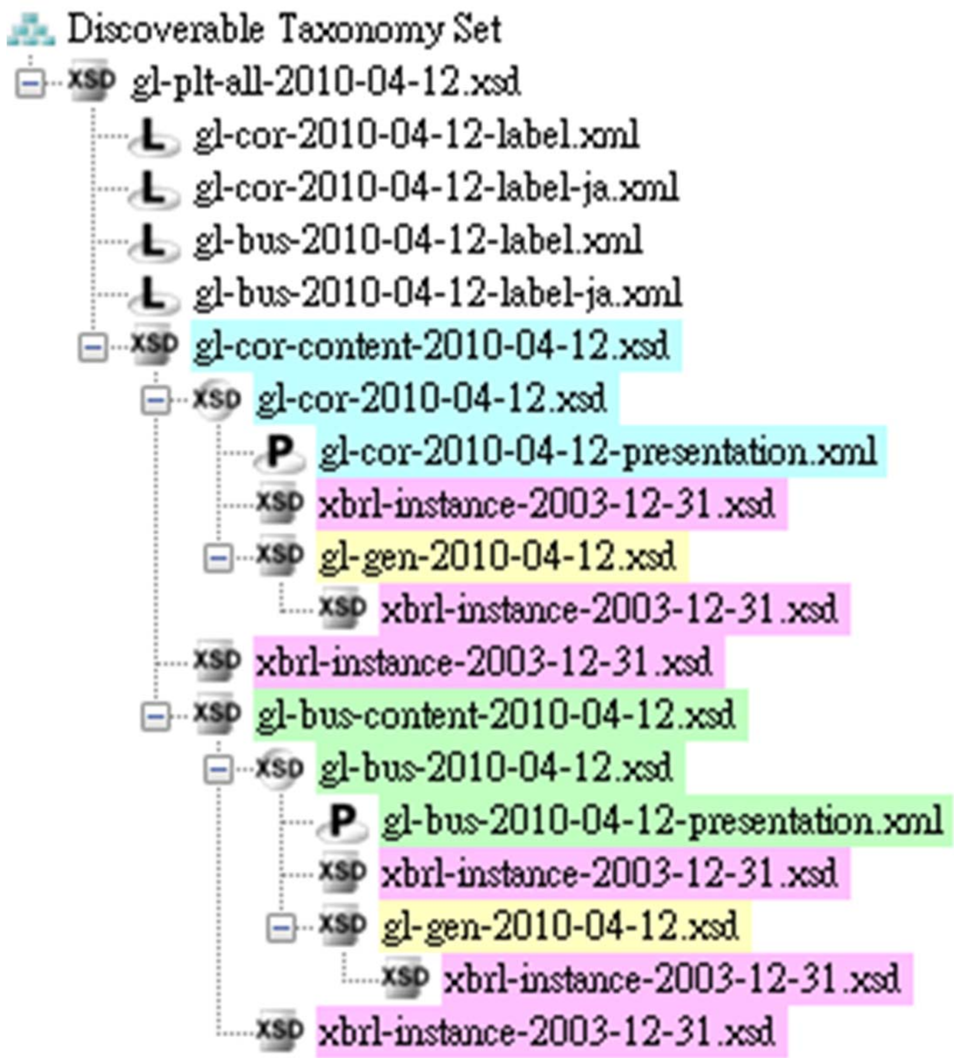
CASE-C套件的DTS



CASE-C-B套件

- 本套件包含五份schema文件：gl-cor-content、gl-bus-content及三份gl-plt 。
- 各文件內容：
  - gl-plt schema文件未做任何元素或型態定義。本文件載入gl-cor-content schema文件，並連結至選定的gl-cor及gl-bus英文或日文label linkbase文件。
  - gl-cor-content schema文件定義86種內容型態。本文件載入gl-bus-content及COR套件的gl-cor等schema文件。
  - gl-bus-content schema文件定義138種內容型態。本文件載入BUS套件的gl-bus schema文件。
    - gl-bus schema文件定義180個元素及9種內容型態。

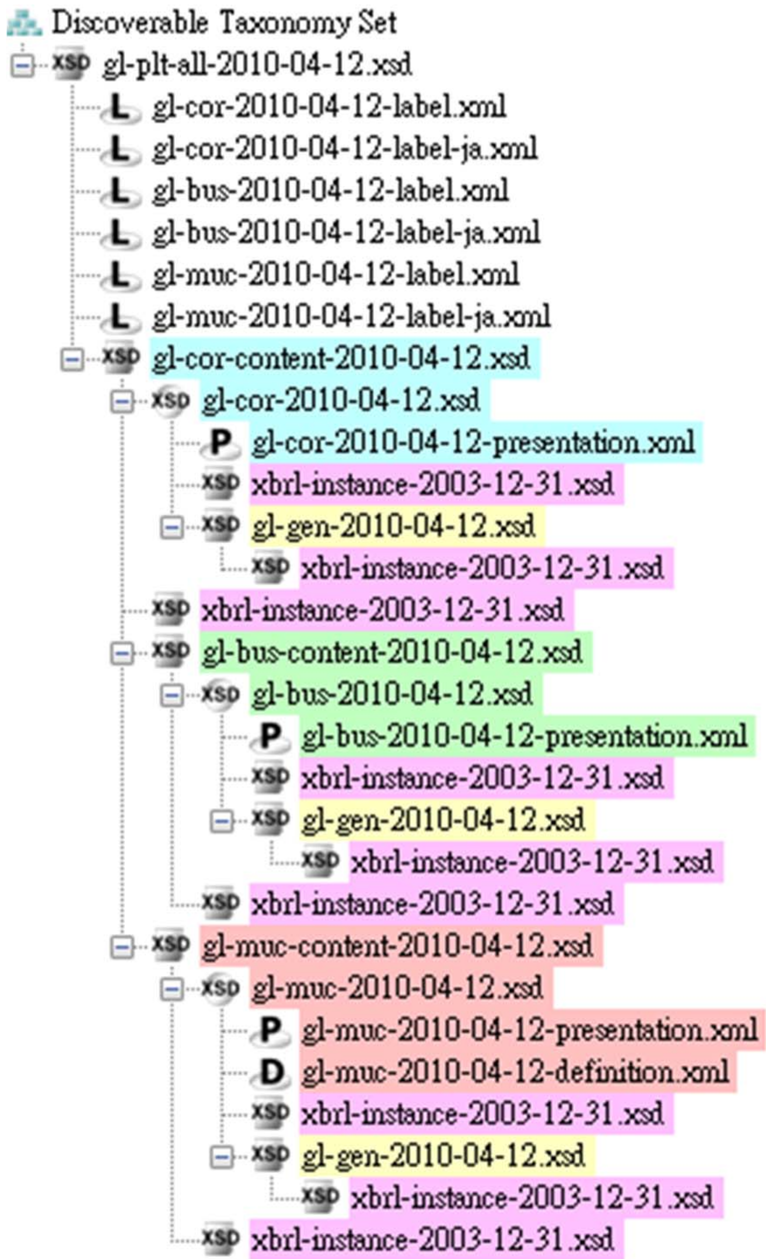
CASE-C-B套件的DTS



CASE-C-B-M套件

- 本套件包含六份schema文件：gl-cor-content、gl-bus-content、gl-muc-content及三份gl-plt。
- 各文件內容：
  - gl-plt schema文件未做任何元素或型態定義。本文件載入gl-cor-content schema文件，並連結至選定的gl-cor、gl-bus及gl-muc英文或日文label linkbase文件。
  - gl-cor-content schema文件定義86種內容型態。本文件載入gl-bus-content、gl-muc-content及COR套件的gl-cor等schema文件。
  - gl-bus-content schema文件定義138種內容型態。本文件載入BUS套件的gl-bus schema文件。
  - gl-muc-content schema文件定義7種內容型態。本文件載入MUC套件的gl-muc schema文件。
    - gl-muc schema文件定義47個元素。
  - MUC套件內含一份definition linkbase文件，以essence-alias arc把分別定義在gl-cor及gl-muc內的taxExchangeRate元素視為同一個概念，並以gl-muc的元素為本名，以gl-cor的元素為別名。

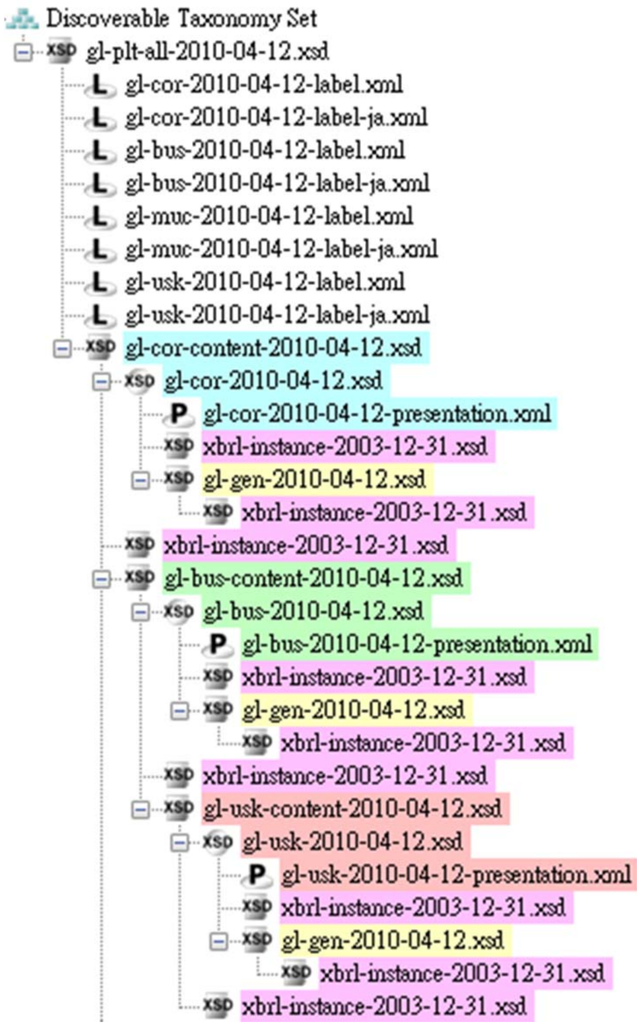
CASE-C-B-M 套件的DTS



CASE-C-B-M-U套件

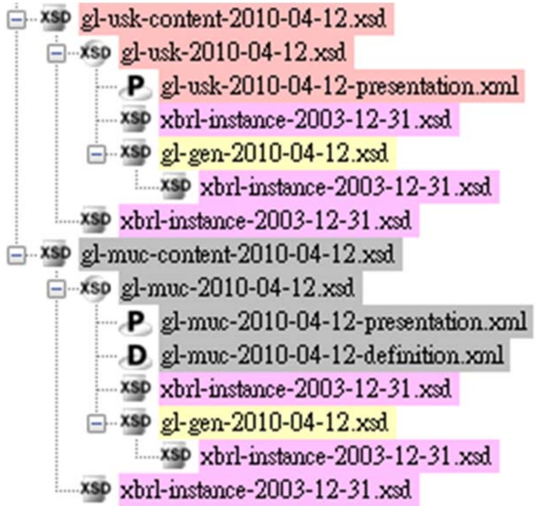
- 本套件包含七份schema文件：gl-cor-content、gl-bus-content、gl-muc-content、gl-usk-content及三份gl-plt。
- 各文件內容：
  - gl-plt schema文件未做任何元素或型態定義。本文件載入gl-cor-content schema文件，並連結至選定的gl-cor、gl-bus、gl-muc及gl-usk英文或日文label linkbase文件。
  - gl-cor-content schema文件定義86種內容型態。本文件載入gl-bus-content、gl-muc-content、gl-usk-content及COR套件的gl-cor等schema文件。
  - gl-bus-content schema文件定義138種內容型態。本文件載入gl-usk-content及BUS套件的gl-bus等schema文件。
  - gl-muc-content schema文件定義7種內容型態。本文件載入MUC套件的gl-muc schema文件。
  - gl-usk-content schema文件定義14種內容型態。本文件載入USK套件的gl-usk schema文件。
    - gl-usk schema文件定義15個元素。

CASE-C-B-M-U套件的DTS



續接右上…

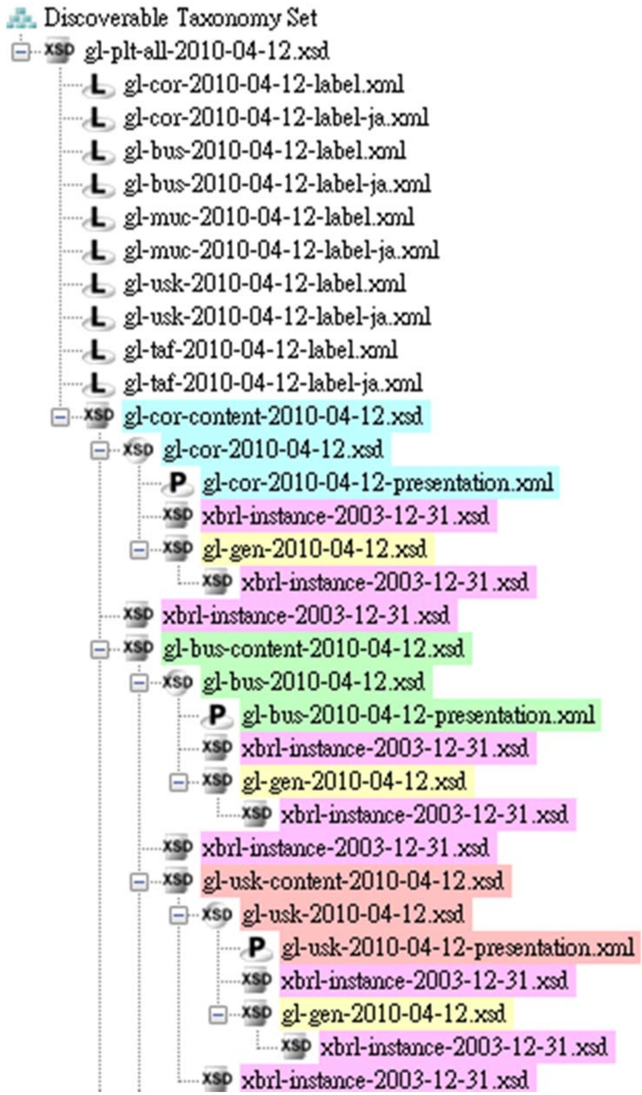
…接自左下



CASE-C-B-M-U-T套件

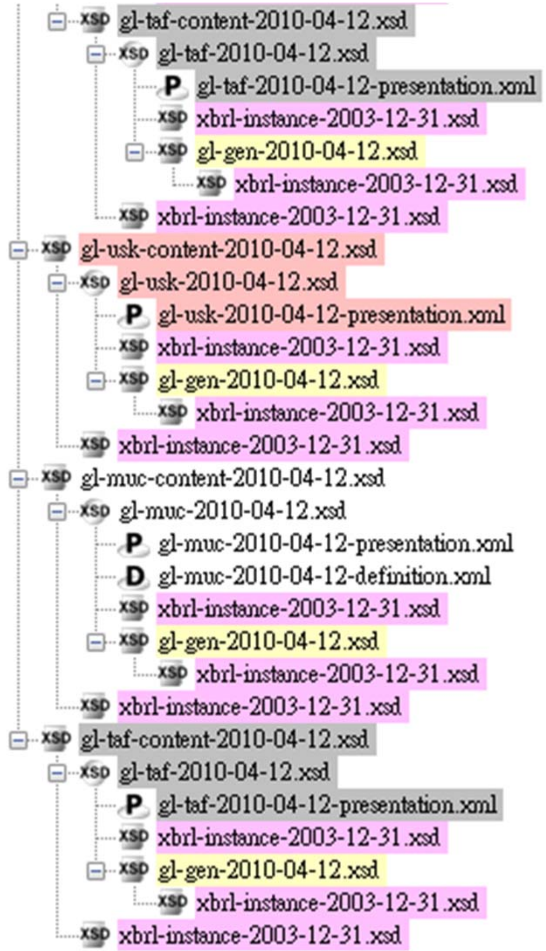
- 本套件包含八份schema文件：gl-cor-content、gl-bus-content、gl-muc-content、gl-usk-content、gl-taf-content及三份gl-plt。
- 各文件內容：
  - gl-plt schema文件未做任何元素或型態定義。本文件載入gl-cor-content schema文件，並連結至選定的gl-cor、gl-bus、gl-muc、gl-usk及gl-taf英文或日文label linkbase文件。
  - gl-cor-content schema文件定義86種內容型態。本文件載入gl-bus-content、gl-muc-content、gl-usk-content、gl-taf-content及COR套件的gl-cor等schema文件。
  - gl-bus-content schema文件定義138種內容型態。本文件載入gl-usk-content、gl-taf-content及BUS套件的gl-bus等schema文件。
  - gl-muc-content schema文件定義7種內容型態。本文件載入MUC套件的gl-muc schema文件。
  - gl-usk-content schema文件定義14種內容型態。本文件載入USK套件的gl-usk schema文件。
  - gl-taf-content schema文件定義10種內容型態。本文件載入TAF套件的gl-taf schema文件。
    - gl-taf schema文件定義10個元素。

CASE-C-B-M-U-T 套件的DTS



續接右上…

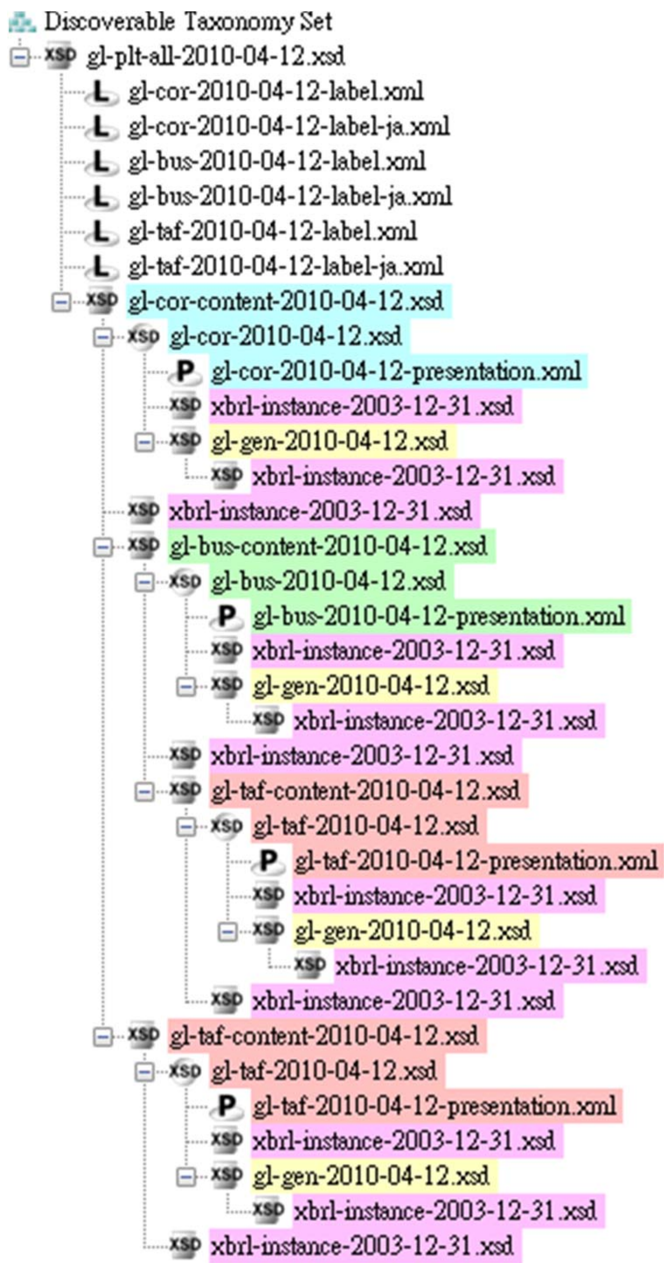
…接自左下



CASE-C-B-T套件

- 本套件包含六份schema文件：gl-cor-content、gl-bus-content、gl-taf-content及三份gl-plt。
- 各文件內容：
  - gl-plt schema文件未做任何元素或型態定義。本文件載入gl-cor-content schema文件，並連結至選定的gl-cor、gl-bus及gl-taf英文或日文label linkbase文件。
  - gl-cor-content schema文件定義86種內容型態。本文件載入gl-bus-content、gl-taf-content及COR套件的gl-cor等schema文件。
  - gl-bus-content schema文件定義138種內容型態。本文件載入gl-taf-content及BUS套件的gl-bus等schema文件。
  - gl-taf-content schema文件定義10種內容型態。本文件載入TAF套件的gl-taf schema文件。

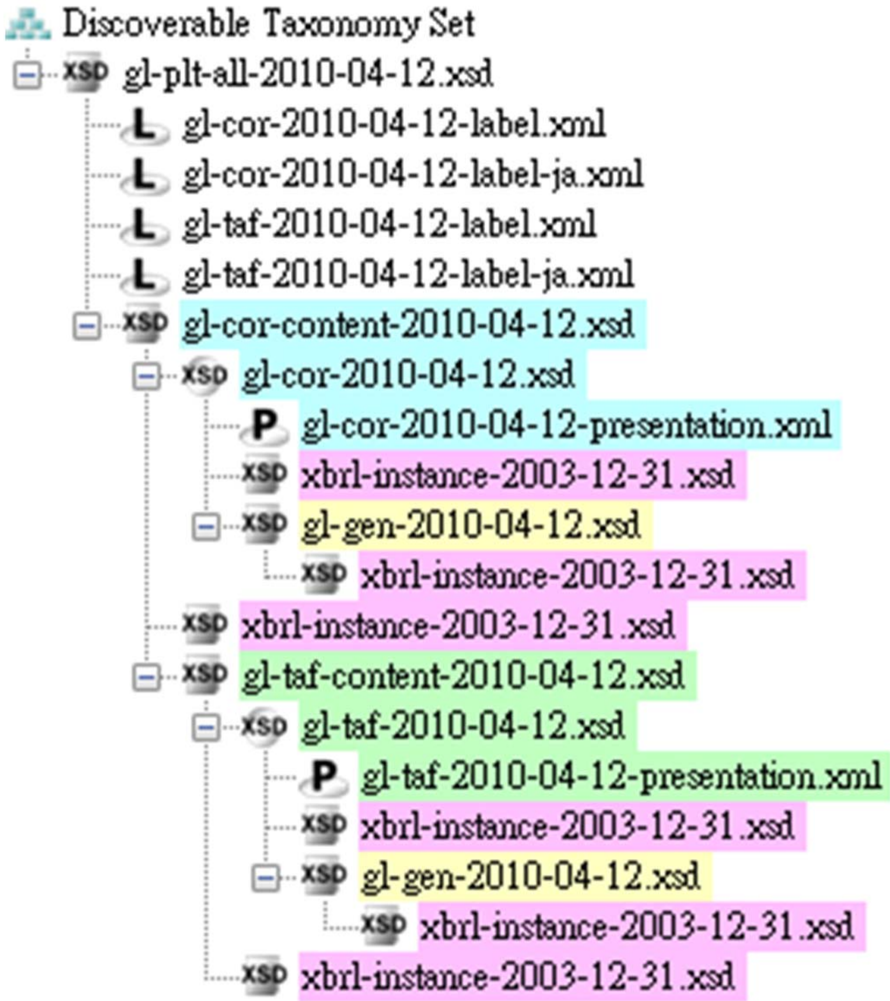
CASE-C-B-T 套件的DTS



CASE-C-T套件

- 本套件包含五份schema文件：gl-cor-content、gl-taf-content及三份gl-plt。
- 各文件內容：
  – gl-plt schema文件未做任何元素或型態定義。本文件載入gl-cor-content schema文件，並連結至選定的gl-cor及gl-taf英文或日文label linkbase文件。
  – gl-cor-content schema文件定義86種內容型態。本文件載入gl-taf-content及COR套件的gl-cor等schema文件。
  – gl-taf-content schema文件定義10種內容型態。本文件載入TAF套件的gl-taf schema文件。

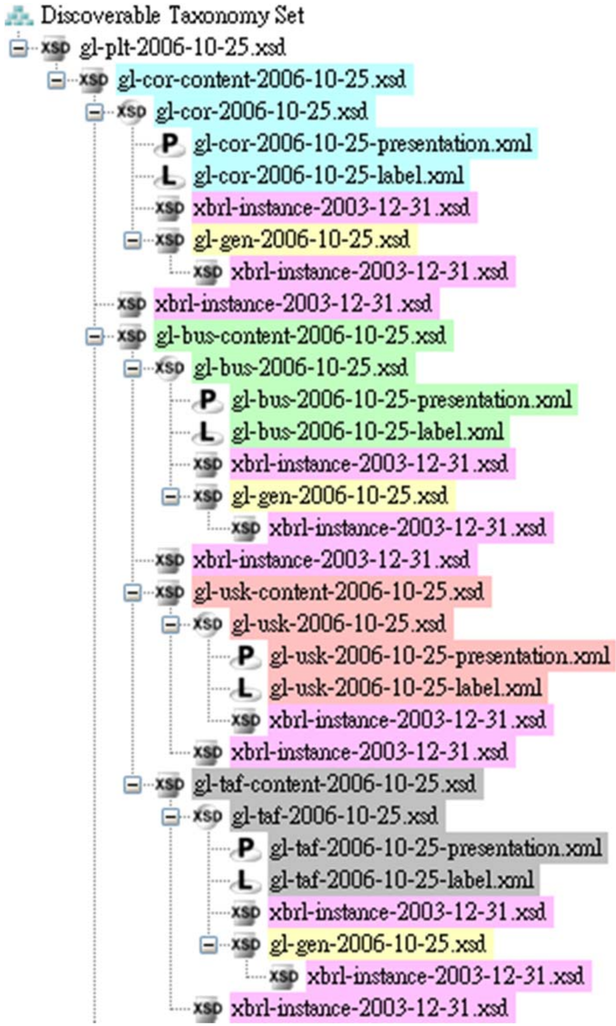
CASE-C-T套件的DTS



最複雜套件：CASE-C-B-M-U-T-S

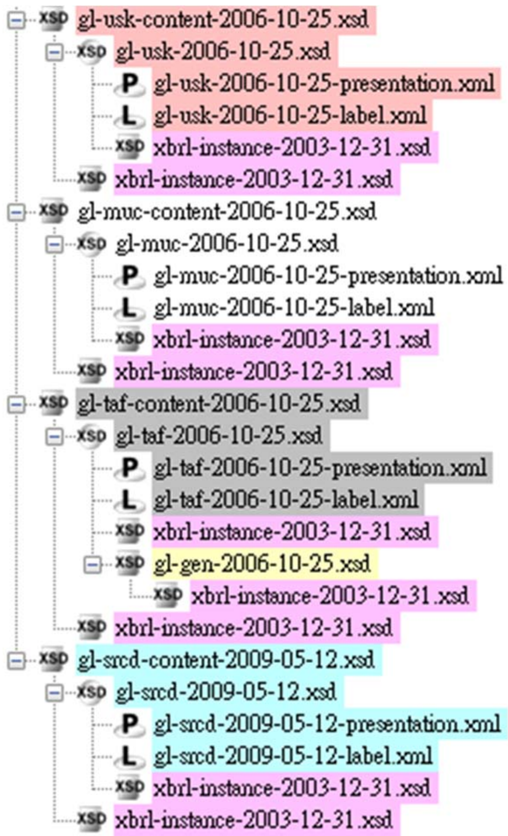
- 在2009+SRCD版本中有本套件，包含七份schema文件：gl-plt、gl-cor-content、gl-bus-content、gl-muc-content、gl-usk-content、gl-taf-content及gl-srcd-content，。
- 各文件內容(2009+SRCD版本)：
  – gl-plt schema文件未做任何元素或型態定義。本文件載入gl-cor-content schema文件。
  – gl-cor-content schema文件定義93種內容型態。本文件載入gl-bus-content、gl-muc-content、gl-usk-content、gl-taf-content、gl-srcd-content及COR套件的gl-cor等schema文件。
  – gl-bus-content schema文件定義144種內容型態。本文件載入gl-usk-content、gl-taf-content及BUS套件的gl-bus等schema文件。
  – gl-muc-content schema文件定義8種內容型態。本文件載入MUC套件的gl-muc schema文件。
  – gl-usk-content schema文件定義15種內容型態。本文件載入USK套件的gl-usk schema文件。
  – gl-taf-content schema文件定義10種內容型態。本文件載入TAF套件的gl-taf schema文件。
  – gl-srcd-content schema文件定義2種內容型態。本文件載入SRCD套件的gl-srcd schema文件。
    - gl-srcd schema文件定義42種內容型態及44個元素。

CASE-C-B-M-U-T-S套件的DTS



續接右上…

…接自左下



gl-cor及gl-cor-content

- gl-cor及gl-cor-content是GL Framework的核心schema文件，也是唯一能在PLT中獨立運作的組合性文件。gl-cor定義119個元素，其中19個是tuple元素，100個是item元素。元素型態則定義於gl-gen及gl-cor-content文件內。
- 核心元素：

  <accountingEntries>：是GL案例文件的內圍根元素，為tuple類型元素，包含以下三種tuple類型子元素：

  - <documentInfo>：包含文件型態、編製者等訊息。gl-cor在此tuple內定義9個子元素。
  - <entityInformation>：包含企業名稱、聯絡人等訊息。 gl-cor在此tuple內並未定義任何子元素，留待gl-bus補充。
  - <entryHeader>：GL文件的核心內容在此表達。gl-cor在此tuple內定義11個item類型子元素及一個tuple類型的<entryDetail>子元素，後者內則包含26個item及tuple類型子元素(若包含tuple內的元素則共有94個元素)。

gl-bus及gl-bus-content

- gl-bus及gl-bus-content是GL Framework的功能別延伸性schema文件，必須由gl-cor-content載入後方能使用。
- gl-bus定義180個與會計資料庫及交易概念有關的進階性元素，其中23個是tuple元素，157個是item元素。
- 元素分佈：180個元素中，6個在<documentInfo>內(COR套件已定義9個子元素)，110個在<entityInformation>內(COR套件並未定義任何子元素) ，64個在<entryHeader>內(COR套件已定義106個子元素) 。

gl-muc及gl-muc-content

- gl-muc及gl-muc-content是GL Framework的功能別延伸性schema文件，必須由gl-cor-content載入後方能使用。
- gl-muc定義47個與多種幣別交易概念有關的元素，其中1個是tuple元素，46個是item元素。
- 元素分佈：47個元素中，1個在<documentInfo>內(COR已定義9個、BUS已定義6個)，46個在<entryHeader>內(COR已定義106個、BUS已定義64個)。
  - 46個位於<entryHeader>內的muc元素，全部都在<entryHeader>的<entryDetail>子元素內，其中，16個是<entryDetail>內的第一層子元素，15個在<entryDetail>的<multicurrencyDetail>子元素內，15個在<entryDetail>的<taxes>子元素內。

gl-usk及gl-usk-content

- gl-usk及gl-usk-content是GL Framework的地域別延伸性schema文件，必須由gl-bus-content及gl-cor-content載入後方能使用。
- gl-usk定義15個常見於英美會計體系、但歐陸各國並不適用的概念元素。這15個元素全部都是item元素。
- 元素分佈：15個元素全部位於<entryHeader>內(COR已定義106個、BUS已定義64個、MUC已定義46個)，提供有關自動化會計分錄(automated accounting entries)及批次成本制(job-costing)的資料標籤。
  - 15個元素中，10個是<entryHeader>內的第一層子元素，5個在<entryHeader>的<entryDetail>的<jobInfo>子元素內。後者是gl-bus的元素，故DTS內gl-usk-content會被gl-cor-content及gl-bus-content重複載入。
  - 自動化會計分錄：在多個期間重複出現、系統會自動產生的分錄，以及期末調整而在次期期初所做的轉回分錄。

gl-taf及gl-taf-content

- gl-taf及gl-taf-content是GL Framework的功能別延伸性schema文件，必須由gl-cor-content載入後方能使用。
- gl-taf定義10個與稅務查核檔案(tax audit file)有關的元素，其中1個是tuple元素，9個是item元素。
- 元素分佈：10個元素全部位於<entryHeader>內(COR已定義106個、BUS已定義64個、MUC已定義46個、USK已定義15個)。
  - 10個元素中，4個是<entryHeader>的<entryDetail>子元素的第一層子元素，另6個在<entryHeader>的 <entryDetail>的<originatingDocumentStructure>子元素內。

gl-srcd及gl-srcd-content

- gl-srcd及gl-srcd-content是GL Framework (2007+SRCD版本)內針對GL與FR文件之連結所制訂的延伸性schema文件，必須由gl-cor-content載入後方能使用。
- gl-srcd定義44個與GL及FR文件之連結有關的元素，其中12個是tuple元素，32個是item元素。
- 元素分佈：44個元素大部分位於<entryHeader>內的<entryDetail>內的<xbrlInfo>子元素內，只有1個tuple(及其4個item子元素)是<entryDetail> 的第一層子元素。
  - 定義於gl-cor schema內的<xbrlInfo>元素原本即設計來處理GL文件與FR文件之間的連結，但原有的內容比較簡略，SRCD套件的功能即在強化此連結機制。

Palette schema文件

- 在XBRL GL 1.1中，負責載入core套件及相關延伸套件的schema文件稱為palette (調色盤) schema文件。GL 1.1的“XBRLGL.xsd”就是一份palette schema文件，它負責載入任意組合的GL 1.1套件。
- 在XBRL GL Framework中，PLT套件的每個次套件中都有gl-plt schema文件，但此文件僅負責連結英、日文label linkbase及載入gl-cor-content文件，後者再負責載入必要的延伸性套件，故調色盤的功能實際上已由gl-cor-content文件取代。

GL提供的審計軌跡

- Audit-trail up: GL與FR的連結
  - 為了讓GL案例文件所表達的內容可以「drill-up」至財務報告中的特定科目，<entryDetail>內可選用一個或多個tuple類型的<xbrlInfo>元素，透過此tuple元素的<xbrlInclude>、<summaryReportingElement>及<detailMatchingElement>三種子元素，即可對映至適當的財報科目。此外，SRCD套件定義44種元素可供進一步處理GL與FR之間的連結。
- Audit-trail down: 連結至憑證
  - <entryDetail>內可包含<documentType>等多種描述原始憑證及記帳憑證的子元素，讓GL案例文件的內容可連結至相關憑證。

XBRL GL Framework案例文件結構 (共2頁)

- GL案例文件需遵循以下的文件結構：

<xbrl>

<schemaRef/> 本元素指向特定plt schema文件，為DTS找尋程序的起點

<context>

提供項目元素所需的背景資訊，但在GL文件中屬於聊備一格的內容

</context>

<unit>

提供數值項目元素所需的單位資訊，但在GL文件中屬於聊備一格的內容

</unit>

<accountingEntries> 是GL案例文件的內圍根元素，GL的實際內容表達於此

<documentInfo>

有關文件類別、編號、編製者、文件內涵之說明等資訊

</documentInfo>

<entityInformation>

有關企業編號、名稱、聯絡人、電話、會計師、企業網頁、企業地址等資訊

</entityInformation>

<entryHeader>

GL案例文件的核心內容在此表達，通常以多個<entryDetail>子元素來呈現

</entryHeader>

</accountingEntries>

</xbrl>

- GLIS 1.0在規則2.1.9及Table 2中，訂定GL案例文件的元素表達順序，根據該順序，context及unit元素應位在accountingEntries元素(facts)之前。
- linkbaseRef、roleRef、arcroleref及footnote並不常出現在案例文件內，因此前頁結構說明未包含這四類元素。

Table 2. Recommended ordering of XBRL instance components.

Elements	Explanation
schemaRef	schemaRef elements MUST be the first children of the xbrl element within an XBRL instance ([XBRL] 4.2).
linkbaseRef	linkbaseRefs MUST appear after the schemaRefs ([XBRL] 4.3).
roleRef	roleRefs MUST appear after the linkbaseRefs ([XBRL] 4.4).
arcroleRef	arcroleRefs MUST appear after the roleRefs ([XBRL] 4.5).
context	Contexts SHOULD appear after arcroleRefs.
unit	Units SHOULD appear after the contexts.
Facts	Facts SHOULD appear after the units.
Footnotes	Footnotes SHOULD appear after the facts.

Context 與 Unit 元素

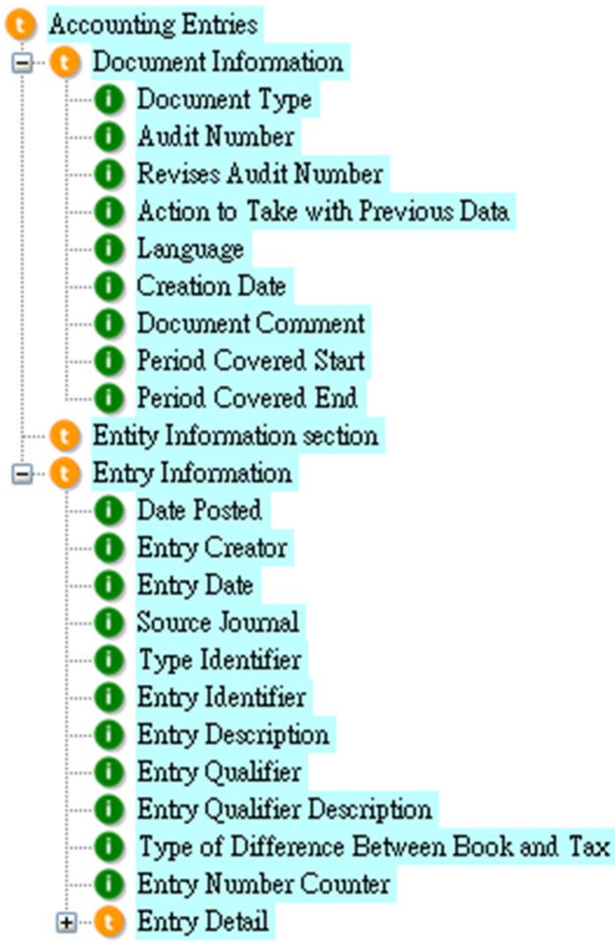
- GL Framework遵循XBRL 2.1規格，案例文件中必須提供context與unit元素讓項目元素參照。
- 但context與unit元素的內容其實是GL案例文件的事實值(facts)，已包含在<accountingEntries>的各項子元素內，故GL案例文件的context與unit元素僅屬聊備一格。
- GLIS 1.0規定：
  - 每個<accountingEntries>的內容只能對應一個context元素。
  - Context元素最好不要包含segment及scenario子元素。
  - Context元素的period子元素，必須包含instant子元素，並以案例文件的編製日期做為instant元素的值。因此，GL分類標準中的item類型元素，在定義時必須將periodType屬性設定為”instant”。
  - Unit元素的內容最好能與數值項目所在之tuple元素內的其他單位相關元素的內容包持一致。
  - 案例文件<u>不應包含</u>單純由比例因子產生的多個unit (例如千元、元、角)，但<u>可包含</u>不同計量單位下的多個unit (例如美元、人民幣)。

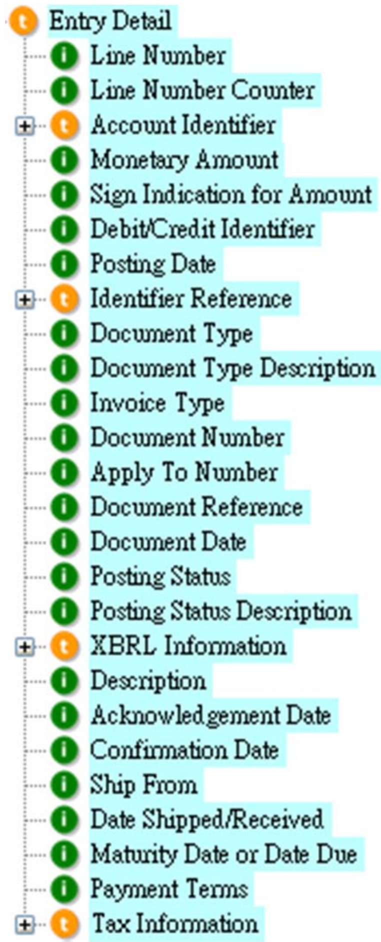
<accountingEntries>

- <accountingEntries>是GL案例文件的內圍根元素，為tuple類型元素。一份GL案例文件可包含一個或多個此tuple元素，該元素內則包含以下三種選用(minOccurs=“0”)的tuple類型子元素：
  - <documentInfo>：包含文件型態、編製者等訊息。一個<accountingEntries>最多只能有一個<documentInfo>元素。
  - <entityInformation>：包含企業名稱、聯絡人等訊息。一個<accountingEntries>最多只能有一個<entityInformation>元素。
  - <entryHeader>：GL文件的核心內容在此表達，通常以多個<entryDetail>子元素(亦為tuple類型)來呈現。一個<accountingEntries>可以有任意多個<entryHeader>元素。

標籤及表達結構

- <accountingEntries>在CASE-C內的標籤及表達結構如右圖。
- 「Entry Information」是<entryHeader>的標籤。其餘標籤與對應的元素名稱大抵相同。





<documentInfo> (共3頁)

- <documentInfo>是<accountingEntries>內的第一種tuple類型選用子元素(可以不選用、但最多一個)。
- GL Framework在<documentInfo>內設計了9~16種item類型子元素(視選用的套件多寡而定)，其中，僅有<entriesType>為必要子元素，其餘15種均為選用子元素。
  - <entriesType>針對GL的文件類別定義了account等9種列舉值。
  - <entriesType>定義在gl-cor schema內，但其內容模型則定義在gl-gen schema內，因此無法透過延伸方式增加GL的文件類別。
- 在包含SRCD套件的結構中， <documentInfo>內另有一個tuple類型的<summaryReportingTaxonomies>元素，內含5種item類型子元素。

- <documentInfo>內的16種item類型子元素：
  - <entriesType>：唯一必要子元素。
  - <uniqueID>
  - <revisedUniqueID>
  - <revisedUniqueIDAction>
  - <language>
  - <creationDate>
  - <creator>
  - <entriesComment>
  - <periodCoveredStart>
  - <periodCoveredEnd>

  續次頁..

接前頁

- <periodCount>
- <periodUnit>
- <periodUnitDescription>
- <sourceApplication>
- <targetApplication>
- <defaultCurrency>

＊ 在包含SRCD套件的結構中，另有一個tuple類型子元素 <summaryReportingTaxonomies>，內含以下五種item類型子元素：

- <summaryReportingTaxonomyID>
- <summaryReportingTaxonomyNamespace>
- <summaryReportingTaxonomySchemaRefHref>
- <summaryReportingTaxonomyHeader>
- <summaryReportingTaxonomyDescription>

<entityInformation> (共2頁)

- <entityInformation>是<accountingEntries>內的第二種tuple類型選用子元素(可以不選用、但最多一個) 。
- GL Framework在<entityInformation>內設計0或15種子元素，前者是在選用的套件組合內未含BUS套件時適用，後者則是在包含BUS套件時適用。
  - 這15種子元素均為選用子元素，定義在BUS schema內。
  - 15種子元素中，10種為tuple類型，5種為item類型。
- <entityInformation>內的15種子元素：
  - <entityPhoneNumber>：tuple類型。
  - <entityFaxNumberStructure>：tuple類型。
  - <entityEmailAddressStructure>：tuple類型。
  - <organizationAccountingMethodPurposeDefault>

續次頁..

接前頁

- <organizationAccountingMethodPurposeDefaultDescription>
- <organizationIdentifiers>：tuple類型。
- <organizationAddress>：tuple類型。
- <entityWebSite>：tuple類型。
- <contactInformation>：tuple類型。
- <businessDescription>
- <fiscalYearStart>
- <fiscalYearEnd>
- <organizationAccountingMethodStructure>：tuple類型。
- <accountantInformation>：tuple類型。
- <reportingCalendar>：tuple類型。

<entryHeader> (共4頁)

- <entryHeader>是<accountingEntries>內的第三種tuple類型選用子元素(可以不選用、亦可選用多個)。此元素的英文標籤為”Entry Information”，所包含的<entryDetail>子元素為GL文件的核心內容所在。
- GL Framework在<entryHeader>內設計了12~38種子元素(視選用的套件多寡而定)，均為選用子元素。 38種子元素中，僅<entryDetail>為tuple類型，其餘均為item類型。且除了<entryDetail>可出現多次外，其餘子元素最多僅能在<entryHeader>內出現一次。
- 在表達日記簿分錄內容時，每個<entryHeader>包含一筆分錄，分錄內的每個借、貸方科目各用一個<entryDetail>表達。

- <entryHeader>內的38種子元素：
  - <postedDate>
  - <enteredBy>
  - <enteredByModified>
  - <enteredDate>
  - <entryResponsiblePerson>
  - <sourceJournalID>
  - <sourceJournalDescription>
  - <entryType>
  - <entryOrigin>
  - <entryNumber>
  - <entryComment>
  - <qualifierEntry>
  - <qualifierEntryDescription>
  - <postingCode>
  - <batchID>
  - <batchDescription>

  續次頁..

接前頁

- <numberOfEntries>
- <totalDebit>
- <totalCredit>
- <bookTaxDifference>
- <eliminationCode>
- <budgetScenarioPeriodStart>
- <budgetScenarioPeriodEnd>
- <budgetScenarioText>
- <budgetScenario>
- <budgetAllocationCode>
- <reversingStdID>
- <recurringStdDescription>
- <frequencyInterval>
- <frequencyUnit>
- <repetitionsRemaining>
- <nextDateRepeat>

續次頁..

接前頁

- <lastDateRepeat>
- <endDateRepeatingEntry>
- <reverse>
- <reversingDate>
- <entryNumberCounter>
- <entryDetail>：唯一tuple類型子元素。

<entryDetail> (共5頁)

- <entryDetail>位於<entryHeader>內，是GL案例文件的核心內容所在，為tuple類型。
- GL Framework在<entryDetail>內設計了26~55種子元素(視選用的套件多寡而定)，均為選用子元素。55種子元素中，9種為tuple類型，其餘為item類型。
  - 在2007+SRCD版本中，再多出一個tuple類型子元素，使得<entryDetail>內最多有56種子元素，
- GL Framework共有9種案例文件類別(由<documentInfo>內的<entriesType>決定)，每種案例文件可視實際需要選用<entryDetail>內的子元素。其中最重要的子元素是<account>及<amount>，前者用以表達科目名稱、編號等資訊，為tuple類型；後者表達科目金額，為貨幣項目型態。

- <entryDetail>內的56種子元素：
  - <lineNumber>
  - <lineNumberCounter>
  - <account>：tuple類型。
  - <amount>
  - <amountCurrency>
  - <amountOriginalExchangeRateDate>
  - <amountOriginalAmount>
  - <amountOriginalCurrency>
  - <amountOriginalExchangeRate>
  - <amountOriginalExchangeRateSource>
  - <amountOriginalExchangeRateComment>
  - <amountOriginalTriangulationAmount>
  - <amountOriginalTriangulationCurrency>
  - <amountOriginalTriangulationExchangeRate>
  - <amountOriginalTriangulationExchangeRateSource>
  - <amountOriginalTriangulationExchangeRateType>

續次頁..

接前頁
- <originalTriangulationExchangeRate>
- <originalExchangeRateTriangulationSource>
- <originalExchangeRateTriangulationType>
- <signOfAccount>
- <debitCreditCode>
- <postingDate>
- <amountMemo>
- <allocationCode>
- <multicurrencyDatail>：tuple類型。
- <identifierReference>：tuple類型。
- <documentType>
- <documentTypeDescription>
- <invoiceType>
- <documentNumber>
- <documentApplyToNumber>
- <documentReference>

續次頁..

接前頁

- <documentDate>
- <documentReceivedDate>
- <documentChargeReimb>
- <documentLocation>
- <paymentMethod>
- <postingStatus>
- <postingStatusDescription>
- <xbrlInfo>：tuple類型。
- <detailComment>
- <dateAcknowledged>
- <confirmedDate>
- <shipFrom>
- <shipReceivedDate>
- <maturityDate>
- <terms>
- <measurable>：tuple類型。

續次頁..

接前頁

- <jobInfo>：tuple類型。
- <depreciationMortgage>：tuple類型。
- <taxes>：tuple類型。
- <tickingField>
- <documentRemainingBalance>
- <uniqueConsignmentReference>
- <originatingDocumentStructure>：tuple類型。
- <richTextComment>：tuple類型。

XBRL GL 案例文件類型

- 根據gl-cor schema及gl-gen schema兩份文件之規範，GL文件共有9種，以<documentInfo>的<entriesType>子元素來顯示其類型，後者也是前者的多種子元素中唯一的必要子元素。
  - <entriesType>的英文標籤為”Document Type”。
- <entriesType>元素有以下九種列舉值，案例文件編製者需從中選擇一種做為GL案例文件的類型：
  - account, balance, entries, journal, ledger, assets, trialbalance, taxtables, other
- 因<entriesType>的內容是由gl-gen定義的，按規定其內容模型不能透過延伸來增加，故不屬於前述類型的案例只能選擇 ”other”。

GL 案例文件類型 1：account

- account：此文件表達會計科目表(Chart of accounts, COA)中各會計科目的編碼及名稱。
  - 結構：每個<account>表達一個科目，此外，因<account>元素可以在<entryDetail>內重複出現，所以可以用單一<entryDetail>表達完整的會計科目表

```
<entryDetail>
  <account>
    <accountMainID/> 會計科目編碼
    <accountMainDescription/> 會計科目名稱
    - - - -
  </account>
  <account>
    <accountMainID/> 會計科目編碼
    <accountMainDescription/> 會計科目名稱
    - - - -
  </account>
  - - - -
</entryDetail>
```

GL 案例文件類型 2：balance

- balance：表達一個或多個科目在特定日期的餘額。
  - 結構：雖然<account>可以在<entryDetail>內重複出現，但用來表達借貸方及金額的<debitCredtiCode>及<amount>只能出現一次，所以每個<entryDetail>只能表達一個科目的餘額，若要表達特定日期所有科目的餘額，整份文件會比較龐大。

```
<entryDetail>
    <account/> 特定科目的敘述性內容
    <debitCreditCode/> 借餘或貸餘
    <amount/> 金額
</entryDetail>
<entryDetail>
    <account/> 特定科目的敘述性內容
    <debitCreditCode/> 借餘或貸餘
    <amount/> 金額
</entryDetail>
- - - -
```

GL 案例文件類型 3：entries

- entries：對特定會計分錄做完整的借貸表達。
  - 結構：通常由一個<entryHeader>元素代表一個特定會計分錄，內含數個彼此借貸平衡的<entryDetail>元素。
  - 例如：借：應收帳款 200,000 貸：銷貨收入 200,000

```
<entryHeader>
    <entryDetail>
        <account>
            <accountMainID>1141</accountMainID>
            <accountMainDescription>應收帳款</accountMainDescription>
        </account>
        <debitCreditCode>d</debitCreditCode>
        <amount>200000</amount>
    </entryDetail>
    <entryDetail>
        <account>
            <accountMainID>4111</accountMainID>
            <accountMainDescription>銷貨收入</accountMainDescription>
        </account>
        <debitCreditCode>c</debitCreditCode>
        <amount>200000</amount>
    </entryDetail>
<entryHeader/>
```

GL 案例文件類型 4：journal

- journal：將一段期間的會計交易如同日記簿的形式依序記錄，每一個<entryHeader>元素代表一個借貸平衡的會計分錄。例：

```
<entryHeader>
    <entryDetail>
        <account/>
        <debitCreditCode/>
        <amount/>
    </entryDetail>
    - - - -
<entryHeader/>
<entryHeader>
    <entryDetail>
        <account/>
        <debitCreditCode/>
        <amount/>
    </entryDetail>
    - - - -
<entryHeader/>
- - - -
```

第一筆會計分錄內容

第二筆會計分錄內容

第三、四.....筆會計分錄

GL 案例文件類型 5：ledger

- ledger：對特定科目在一段期間的增減做完整的紀錄，類似分類帳的功能。例：

```
<entryHeader>
    <entryDetail>
        <account>
            <accountMainID>1141</accountMainID>
            <accountMainDescription>應收帳款</accountMainDescription>
        </account>
        <debitCreditCode>d</debitCreditCode>
        <amount>200000</amount>
    </entryDetail>
    <entryDetail>
        <account>
            <accountMainID>1141</accountMainID>
            <accountMainDescription>應收帳款</accountMainDescription>
        </account>
        <debitCreditCode>c</debitCreditCode>
        <amount>120000</amount>
    </entryDetail>
    - - - -
<entryHeader/>
```

GL 案例文件類型 6：assets

- assets：表達資產科目在一段期間的部分增減變動，但不若ledger完整。例如，應收特定客戶、應付特定供應商帳款分析、特定存貨分析、特定固定資產分析。

GL 案例文件類型 7：trialbalance

- trialbalance：將各科目在一段期間內的借記總額與貸記總額分別列示，所有科目的借記加總應等於貸記加總，以符和試算表的要求。
  - 結構：每個<entryHeader>元素代表一個會計科目，內含兩個<entryDetail>元素，分別表達該科目在一段期間內的借記總額與貸記總額。例：

```
<entryHeader>
    <entryDetail>
        <account>
            <accountMainDescription>現金</accountMainDescription>
        </account>
        <debitCreditCode>d</debitCreditCode>
        <amount>25376814</amount>
    </entryDetail>
    <entryDetail>
        <account>
            <accountMainDescription>現金</accountMainDescription>
        </account>
        <debitCreditCode>c</debitCreditCode>
        <amount>18973653</amount>
    </entryDetail>
<entryHeader/>
- - - -
```

GL 案例文件類型：taxtables, other

8. taxtables：提供稅務查核所需的相關內容。

9. other：以上8種類別之外的案例文件內容。

XBRL GL 分錄類型

- 根據gl-cor schema及gl-gen schema兩份文件之規範，由<entryHeader>表達的分錄可分成13種，以<entryHeader>內的<entryType>子元素來顯示其類型。
- <entryType>元素的13種列舉值：
  - adjusting, budget, comparative, external-accountant, standard, passed-adjusting, eliminating, proposed, recurring, reclassifying, simulated, tax, other
  - 例如，一般分錄為standard，調整分錄為adjusting，重分類分錄為recalssifying..等。

※ <entriesType>為<documentInfo>內的必要子元素，表達案例文件類型，其英文標籤為"Document Type"；<entryType>為<entryHeader>內的選用子元素，表達分錄類型，其英文標籤為"Type Identifier"。

XBRL GL 帳戶訊息 (共2頁)

- GL案例文件的帳戶資訊表達在<account>內，此元素是<entryDetail>內的核心子元素，為tuple類型。<account>的英文標籤為”Account Identifier”。
- <account>內最多可有15種選用子元素：
  – <accountMainID> 會計科目編碼
  – <accountMainDescription> 會計科目名稱
  – <mainAccountType> 會計科目類別：資產、負債、權益、收入、利得、費用、損失、貢獻給權益、分配自權益、綜合淨利等十種(英文)列舉值中擇一
  – <mainAccountTypeDescription> 科目類別說明
  – <parentAccountMainID> 父帳戶科目編碼
  – <accountPurposeCode> 帳戶使用目的：合併、GAAP、報稅、地域別規定…等
  – <accountPurposeDescription> 使用目的說明
  – <accountType> 帳戶類型：從account、bank、employee、customer、job、vendor、fixed-asset、statistical、other等九種列舉值中擇一。

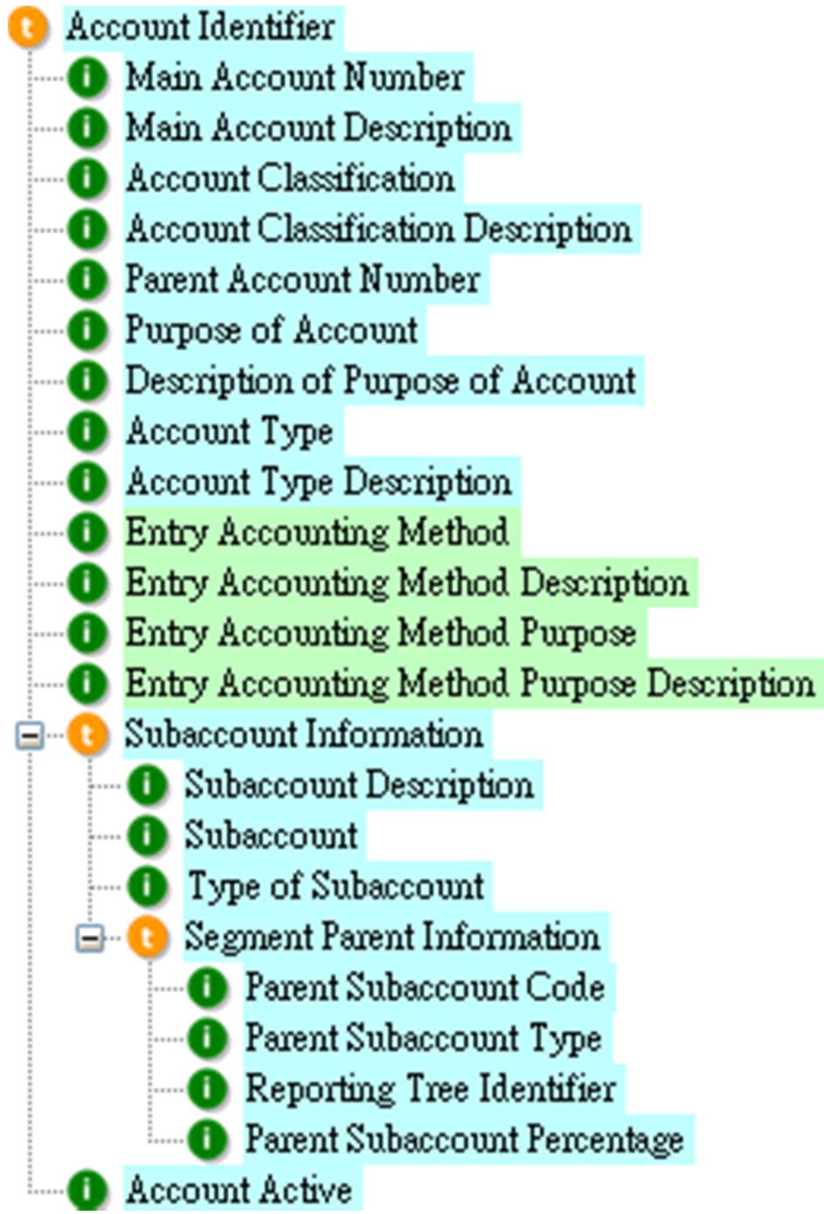
…續次頁

接前頁

- <accountTypeDescription> 帳戶類型說明
- <entryAccountingMethod> 分錄會計方法：從accrual、cash、modified cash、modified accrual、encumbrance、special methods、hybrid methods、other等八種列舉值中擇一
- <entryAccountingMethodDescription> 分錄會計方法說明
- <entryAccountingMethodPurpose> 分錄會計方法目的：從book、tax、management、statutory、other等五種列舉值中擇一
- <entryAccountingMethodPurposeDescription> 分錄會計方法目的說明
- <accountSub> 明細帳資訊：包含多種描述部門別及分支機構相關科目的子元素
- <accountActive> 為boolean型態

• 上述15種子元素均為選用元素(minOccurs="0")。其中，僅<accountSub>為tuple類型，且可出現多次；其餘為item類型，最多只能出現一次。

<Account>元素的標籤及表達結構



XBRL GL 會計科目類別

- 根據gl-cor schema及gl-gen schema兩份文件之規範，由<account>表達的會計科目可分成11種，以<account>內的<mainAccountType>子元素來顯示其類別，此子元素的英文標籤為”Account Classification”。
- <mainAccountType>元素的11種列舉值：
  - asset, liability, equity, income, gain, expense, loss, contr-to-equity, distr-from-equity, comprehensive-income, other
  - 以上11種列舉值的前10種，即所謂十大會計要素。

XBRL GL 帳戶類型

- 根據gl-cor schema及gl-gen schema兩份文件之規範，由<account>表達的帳戶可分成9種，以<account>內的<accountType>子元素來顯示其類型，此子元素的英文標籤為”Account Type”。
- <accountType>元素的9種列舉值：
  – account, bank, employee, customer, job, vendor, measurable, statistical, other
  – 由上述9種列舉值可知，XBRL GL的<account>元素除了可用來表達會計科目表的帳戶(account)，也能用來表達銀行往來帳(bank)、員工(employee)、客戶(customer)、工作批次(job)、供應商(vendor)、衡量指標(measurable)、統計數據(statistical)等其他功能的帳戶。
  – 因此，XBRL GL在設計上是一個超越會計總帳內涵的資料表達及擷取架構，故英文名稱改為 “global” ledger。

XBRL GL 帳戶目的碼

- 同一帳戶，可能會因目的不同而做不同表達。例如，跨國經營的企業，特定會計科目餘額會因按照不同GAAP (e.g. usgaap、ifrs) 計算而有不同結果；此外，會計帳餘額與報稅時帳外調整餘額亦有不同。
- 根據gl-cor schema及gl-gen schema兩份文件之規範，由<account>表達的同一帳戶可能有9種目的，以<account>內的<accountPurposeCode>子元素來顯示其目的類型，此子元素的英文標籤為”Purpose of Account”。
- <accountPurposeCode>元素的9種列舉值：
  – consolidating, european, ifrs, offsetting, primary, tax, usgaap, japanese, other.

明細帳連結訊息

- GL Framework在<account>內定義一個可選用多次的tuple類型<accountSub>子元素(英文標籤為“Subaccount Information”)，內含以下4種子元素，提供明細帳連結訊息：
  - <accountSubID> 明細帳編號
  - <accountSubDescription> 明細帳說明
  - <accountSubType> 明細帳類型，可填入任何token字串值。GL工作小組在Amazing Account文件中列舉8種值：division, project, branch, profit center, department, business unit, program, fund。
  - <segmentParentTuple> 為tuple類型，內含4種item類型子元素：
    - <parentSubaccountCode>
    - <parentSubaccountType>
    - <reportingTreeIdentifier>
    - <parentSubaccountProportion>

Amazing Account (共2頁)

- “GL”這個名稱讓XBRL GL被大多數人誤認為只是用來處理會計總帳內涵的技術規範，XII的GL工作小組特別準備了“Amazing Account” 說明文件，希望能破除這個誤解。
- GL Framework的<account>概念，遠遠超過一般認定的會計帳戶概念：<account>除了用來表達會計科目或帳戶外，也可用來表達客戶、供應商、員工、存貨細項、個別固定資產、個別銀行帳戶...等。
  - 在GL的設計上，一筆會計分錄由一個<entryHeader>紀錄，若分錄內之借貸方科目共有n個，該<entryHeader>內就有n個<entryDetail>，分別處理單一科目的內容。
  - 一個<entryDetail>內可包含多個<account>，但只能有一個<amount>。但一般會計分錄的個別科目並不會出現「單一金額卻需記錄在兩個帳戶」的情形。

- 跨國經營的企業有時需將交易按不同GAAP或COA進行記錄，但金額通常會跟著調整，與上述多個<account>配上單一<amount>的設計並不相符。
- <account>內含<accountSub>，因此即使需同時記錄在總帳及明細帳內，也不需要兩個<account>。
- 因此單一<entryDetail>內可包含多個<account>並不是針對會計處理之需求而設計的，但卻可讓其他類型之表達具有更大的彈性。
- 例如，可以用一個<entryDetail>來代表某個存貨類別內10樣存貨的內容，每樣存貨用一個<account>來描述，如果該存貨還可細分成子項目，就使用<account>內的<accountSub>來描述。

XBRL GL 憑證訊息

- <entryDetail>內有10種與原始或記帳憑證相關的元素，均為item類型：
  - <documentType>：憑證類型，共有15種列舉值。[註：此元素和<entriesType>元素有相同的英文標籤，都是 “Document Type”]
  - <documentTypeDescription>：憑證類型描述。
  - <invoiceType>：發票類型，從 “ePos” 或 “self-billed” 二擇一。
  - <documentNumber>：憑證編號。
  - <documentApplyToNumber>
  - <documentReference>
  - <documentDate>：文件編製日期。
  - <documentReceivedDate>：文件收到日期。
  - <documentChargeReimb>
  - <documentLocation>：憑證存放地點。

XBRL GL 憑證類型

- 根據gl-cor schema及gl-gen schema兩份文件之規範，與GL文件相關的憑證可分成15種，以<entryDetail>內的<documentType>子元素來顯示其類型。
- < documentType >元素的15種列舉值：
  - check, debit-memo, credit-memo, finance-charge, invoice, order-customer, order-vendor, payment-other, reminder, tegata, voucher, shipment, receipt, manual-adjustment, other.
  - tegata係日文「券」之音譯。

XBRL GL 過帳狀態訊息

- GL Framework在<entryDetail>內定義3種與過帳狀態有關的item類型子元素：
  - <postingDate>：過帳日期。此外，在<entryHeader>內另有<entryDate>描述GL分錄製作日期。
  - <postingStatus>：過帳狀態，有deferred, posted, proposed, simulated, tax, unposted, cancelled, other等8種列舉值。
  - <postingStatusDescription>：過帳狀態描述。

XBRL GL 稅務訊息 (共3頁)

- GL Framework定義多種與稅務處理相關的元素：
  - 在<entryHeader>內：
    - <bookTaxDifference>：帳上與報稅差異類型，有permanent、temporary及none三種列舉值。
  - 在<entryDetail>內：
    - <taxes>：tuple類型，內含9~24種item類型子元素。
    - <tickingField>
    - <documentRemainingBalance>
    - <uniqueConsignmentReference>
    - <originatingDocumentStructure>：tuple類型，內含6種item類型子元素。

    ＊以上除<taxes>定義於COR套件外(9~24種item子元素分別定義於COR及MUC套件)，後面4種元素及其子元素均定義於TAF套件。

- <taxes>內有以下24種選用子元素：
  - <taxAuthority>：稅務機關。
  - <taxTableCode>：報稅表格代碼。
  - <taxDescription>
  - <taxAmount>：稅款金額。
  - <taxBasis>：課稅基礎(稅基)金額
  - <taxExchangeRate>：外幣稅款匯率。在包含MUC套件時，此元素由MUC內的同名元素取代，MUC套件內的definition linkbase把本元素稱為別名(alias)，把MUC內的同名元素稱為本名(essence)。
  - <taxPercentageRate>：稅率。
  - <taxCode>：稅別。
  - <taxCommentExemption>

  ~以上9元素定義於COR套件內，以下15元素定義於MUC套件內~
  - <taxAmountForeignCurrency>：外幣稅款金額。
  - <taxCurrency>：外幣稅款幣別。

  續次頁..

接前頁

- <taxExchangeRateDate>：外幣稅款匯率日期。
- <taxExchangeRate>：外幣稅款匯率(一元本國幣/n元外幣)。
- <taxExchangeRateSource>
- <taxExchangeRateType>
- <taxExchangeRateComment>
- <taxAmountTriangulationCurrency>
- <taxTriangulationCurrency>
- <taxTriangulationExchangeRate>
- <taxTriangulationExchangeRateSource>
- <taxTriangulationExchangeRateType>
- <taxForeignTriangulationExchangeRate>
- <taxForeignTriangulationExchangeRateSource>
- <taxForeignTriangulationExchangeRateType>

GL 與帳簿連結訊息

- 為了讓GL分錄內容能與記載該內容的原始帳簿文件相連結，GL Framework在<entryDetail>內設計2種item類型子元素：
  - <sourceJournalID>：原始帳簿代碼，有cd, cr, fa, gi, gj, im, jc, pj, pl, sj, se, ud, ot等13種列舉值。
  - <sourceJournalDescription>：原始帳簿描述。

GL 與 FR 案例連結訊息

- 為了讓GL分錄內容能與財務報告端案例文件內容相連結，GL Framework在<entryDetail>內設計可重複出現的tuple類型<xbrlInfo>選用子元素。在未包含SRCD的CASE中，此元素內含3種item類型子元素：
  - <xbrlInclude>：用來指出被表達的訊息是期初、期末或期間變動數據，共有 “beginning_balance”、“ending_balance”、“period_change”等三種列舉值。
  - <summaryReportingElement>：指向FR Taxonomy內的特定元素，為QName型態。
  - <detailMatchingElement>：指向GL Taxonomy內的特定元素，為QName型態。
- 在包含SRCD的CASE中，< entryDetail >內額外包含44種item及tuple類型子元素，其中大部分新增元素仍然位於<xbrlInfo>內。

GL案例文件範例

- GL Instance Best Practice：由XBRL GL工作小組在2007/8/1發佈的文件，內含十份有詳細註解的GL案例文件範例。

  文件網址： http://www.xbrl.org/GLWGNotes/XBRL-GL-WGN-Best_Practice_Annotated_Instances-2007-08-01.rtf
- 在新版本GL技術文件中，上述案例文件範例已包含在ids資料夾內。

XBRL GL的應用實例

- Wacoal：一家日本服飾製造商(華歌爾)，在Hitachi協助下成為全球第一個在ERP系統中內建XBRL GL帳務處理系統的公司。
- PCA：一家日本商業會計套裝軟體公司，是全球第一個將XBRL GL設計成標準文件輸出格式的公司。
- CaseWare：在其旗艦軟體產品Working Papers中全面支援XBRL GL的各項功能。
- HUD：美國聯邦政府住宅及都市發展部，使用 DynAccSys公司的XABRA軟體及Fujitsu公司的XWand軟體，建立以XBRL GL為基礎的會計資訊架構。
- Business Objects：在其商業智慧分析軟體中支援XBRL GL文件之分析。

附錄1：FR與GL案例文件結構比較 (共4頁)

範例：某企業本年度的銷貨收入是$2,000,000，銷貨成本是$1,500,000，推銷費用是$150,000，管理費用是$50,000，因此稅前純益是$300,000

用XBRL FR表達

```
<xbrl>
  <schemaRef/>
  <context/>
  <unit/>
  <SalesRevenue>2000000</SalesRevenue>
  <CostOfSale>1500000</CostOfSale>
  <SellingExpenses>150000</SellingExpenses>
  <AdministrativeExpenses>50000</AdministrativeExpenses>
  <EarningsBeforeTax>300000</EarningsBeforeTax>
</xbrl>
```

說明：在FR案例中，科目內容直接填入代表各科目的元素內。各科目的借貸方訊息定義在FR分類標準內。

用XBRL GL表達

```
<xbrl>
  <schemaRef/>
  <context/>
  <unit/>
  <accountingEntries>
    <documentInfo>
      <entriesType>balance</entriesType>
    </documentInfo>
    <entryHeader>
      <entryDetail>
        <account>
          <accountMainID>4111</accountMainID>
          <accountMainDescription>銷貨收入</accountMainDescription>
        </account>
        <debitCreditCode>c</debitCreditCode>
        <amount>2000000</amount>
      </entryDetail>
      續…
```

```xml
<entryDetail>
  <account>
    <accountMainID>5111</accountMainID>
    <accountMainDescription>銷貨成本</accountMainDescription>
  </account>
  <debitCreditCode>d</debitCreditCode>
  <amount>1500000</amount>
</entryDetail>
<entryDetail>
  <account>
    <accountMainID>6150</accountMainID>
    <accountMainDescription>推銷費用</accountMainDescription>
  </account>
  <debitCreditCode>d</debitCreditCode>
  <amount>150000</amount>
</entryDetail>
<entryDetail>
  <account>
    <accountMainID>6250</accountMainID>
    <accountMainDescription>管理費用</accountMainDescription>
  </account>
  <debitCreditCode>d</debitCreditCode>
  <amount>50000</amount>
</entryDetail>
```

續..

```xml
        <entryDetail>
          <account>
            <accountMainID>7911</accountMainID>
            <accountMainDescription>稅前純益</accountMainDescription>
          </account>
          <debitCreditCode>c</debitCreditCode>
          <amount>300000</amount>
        </entryDetail>
      </entryHeader>
    </accountingEntries>
</xbrl>
```

說明：在GL案例中，需在<account>元素內填入各科目名稱，在<debitCreditCode>內填入借貸方，然後在<amount>元素內填入金額。

附錄2：相關網頁連結

- XBRL GL Framework 技術文件檔案

  http://www.xbrl.org/int/gl/2010-04-12/GLFramework+ja-labels-PR-2010-04-12.htm
- Amazing Account：GL工作小組針對<account>元素的功能所做的詳細說明

  http://www.xbrl.org/GLWGNotes/XBRL-GL-WGN-Amazing_Account-2007-07-08.rtf
- Dimensions：GL工作小組針對GL技術是否應改採用dimension規格的明確回應

  http://www.xbrl.org/WGN/Dimensions/WGN-2009-06-03/Dimensions-WGN-WGN-2009-06-03.html