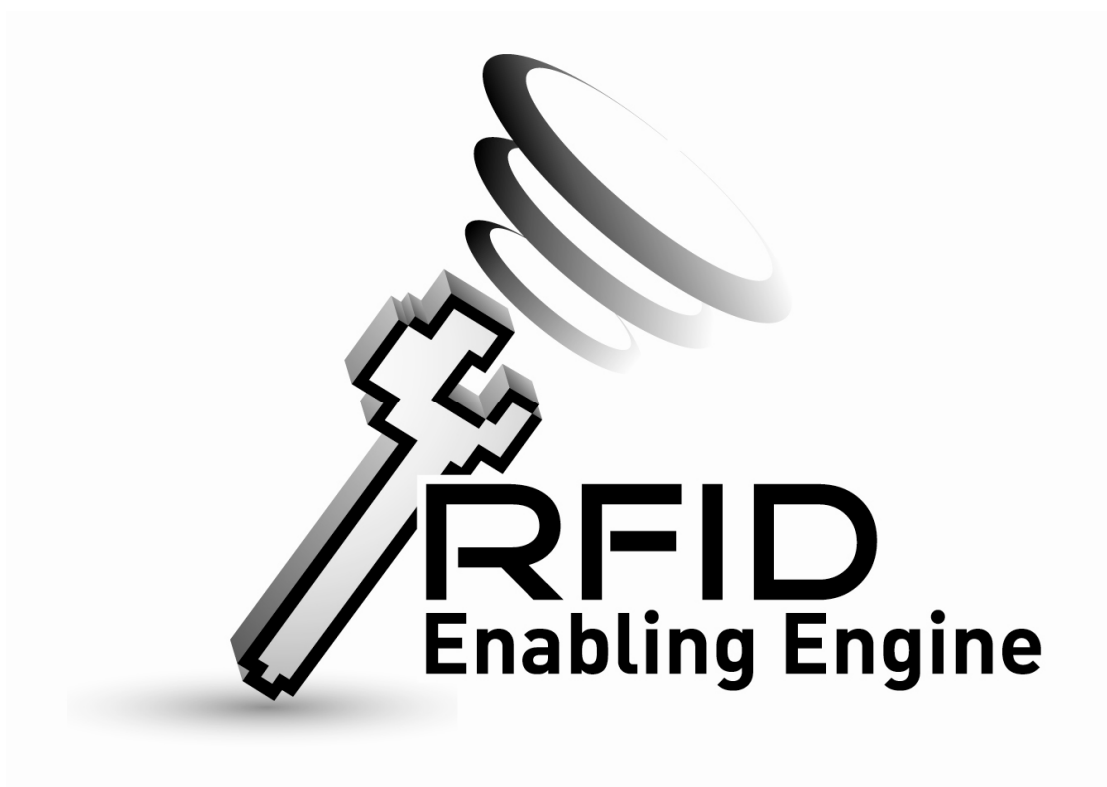

技術規格參考

RFID Enabling Engine 實施 (RFP_RFID_EE_001)



1	摘要	4
2	介紹	5
3	參考	7
4	架構陳述	8
5	外在環境關係	9
6	需求觀點	15
6.1	系統功能	15
6.2	Use Cases	17
7	邏輯觀點	22
7.1	程序設計	22
7.2	設計方式	22
8	進程觀點	41
8.1	RFID Enabling Engine - RFID 硬件溝通介面	41
8.2	RFID Enabling Engine – EPCIS 溝通介面	41
8.3	RFID Enabling Engine - RFID Middleware 溝通介面	41
9	實施觀點	43
9.1	資料庫及程式伺服器考量	43
9.2	Class Diagram	44
9.3	Collaboration Diagram	50
9.4	Package Diagram	60
10	部署觀點	64
10.1	運作環境	64
11	資料觀點	65
11.1	數據庫表 config	65
11.2	數據庫表 epics_datastore	66
11.3	數據庫表 epcis_event	67
11.4	數據庫表 pattern	68
11.5	數據庫表 printorder	69
11.6	數據庫表 rfiditem	70
11.7	數據庫表 taghistory	71
12	系統特性	72
12.1	延伸性	72
12.2	穩定性	72
12.3	多樣性	72
12.4	可攜性	72

附錄 A 資料庫規格	73
附錄 B 用戶介面設計.....	84
附錄 C 製作 JAR 簡單指引.....	93
附錄 D RFID Enabling Engine 及 EPCIS 附加資料.....	94
附錄 E RFID Enabling Engine 及 DATAPLEX 附加資料.....	96

1 摘要

無線射頻識別技術(RFID)是一項可利用無線電波以進行識別及追蹤之技術，應用範疇包括產品、動物及人類。由於其廣泛應用，**香港物流商會**遂贊助此一項目以研發一開放源代碼之 RFID Enabling Engine 程式以幫助及完善整個物流行業之產品追蹤模式。

此文件為一技術參考文件以解釋如何開發 RFID Enabling Engine 及其相關設計。RFID Enabling Engine 主要功能如下：

- 標籤打印功能
- 激活標籤功能
- 關聯標籤功能
- 讀取標籤功能
- 分離標籤功能
- 核對標籤功能

用戶可於 **GS1 Hong Kong** 下載此 RFID Enabling Engine。RFID Enabling Engine 為一 J2EE 程式並容許下載用戶修改或直接執行。RFID Enabling Engine 支援 **GS1 EPCglobal** 之 EPC 及 EPCIS 格式。

用戶可應用 RFID Enabling Engine 上述所列之功能。根據情況，用戶可透過 XML 格式與其他程式溝通。如有不同 RFID 硬件(例: RFID 讀取器)需支援，RFID Enabling Engine 亦確保能透過 RFID Middleware 進行溝通。

2 介紹

無線射頻識別技術(RFID)是一項可利用無線電波以進行識別及追蹤之技術，應用範疇包括產品、動物及人類。由於其廣泛應用，**香港物流商會**遂贊助此一項目以研發一開放源代碼之 RFID Enabling Engine 程式以幫助及完善整個物流行業之產品追蹤模式。

此文件為一技術參考文件以解釋如何開發 RFID Enabling Engine 及其相關設計。RFID Enabling Engine 之總體功能及結構將會於本文件各部份予以詳細描述，包括：

第三部份提供本文件所曾參考之文件及標準規範。用戶可查閱列明之參考文件以作為本文件之進一步之資料。

第四部份介紹本文件之結構展示模式。本文件將會以觀點模組形式詳細剖析 RFID Enabling Engine 之結構，包括以不同角度及層面分析。

第五部份以宏觀角度描述 RFID Enabling Engine 與其他程式之互動關係。其他程式如 RFID Middleware 或用戶程式。

第六部份以 Use Case 模型描述 RFID Enabling Engine 所提供之功能。

第七部份提供 RFID Enabling Engine 之邏輯觀點，詳細解構設計概念及系統架構。

第八部份提供 RFID Enabling Engine 之進程觀點，詳細描述系統不同之資料交換介面，包括 RFID Enabling Engine 與 Middleware 介面、RFID Enabling Engine 與其他程式介面及 RFID Enabling Engine 與硬件介面。

第九部份提供 RFID Enabling Engine 之開發觀點，包括 Class Diagram, Collaboration Diagram 等。

第十部份提供 RFID Enabling Engine 之實施觀點，包括安裝環境要求及安裝 RFID Enabling Engine 之基本步驟。

第十一部分提供 RFID Enabling Engine 之數據庫架構。(數據庫規格將會於附錄 A 列明)

第十二部份將描述 RFID Enabling Engine 一些特性。概念性的用戶介面範例將會於附錄 B 顯示)

3 參考

補充資料及文件將詳列如下以供用戶參考。根據以下參考文件，用戶可進一步獲得 RFID Enabling Engine 如何建立及其建立之原因:

文件	版本	日期	機構
The EPCglobal Architecture Framework	1.3	19 / 3 / 2009	EPCglobal Inc.
EPC Information Services (EPCIS) Version 1.0.1 Specification	1.0.1	21 / 9 / 2007	EPCglobal Inc.
GS1 GTIN Check Digit Calculator Algorithm (Web)	-	-	GS1
RFID Enabling Engine Installation Guide	1.0		GS1 HK
RFID Enabling Engine User Manual	1.0		GS1 HK

4 架構陳述

本文件以不同之觀點模組陳述 RFID Enabling Engine 之技術規格，包括：

需求觀點:	描述功能性及非功能性之 RFID Enabling Engine 需求，並以 Use Case 模型顯示
邏輯觀點:	描述系統設計的 Object Oriented 模型及相關 Package
進程觀點:	描述不同個體與 RFID Enabling Engine 之互動介面
實施觀點:	以 Class Library 形式描述 RFID Enabling Engine 之功能及相關 Library
部署觀點:	描述 RFID Enabling Engine 與硬件環境之關係
資料觀點:	描述 RFID Enabling Engine 之資料庫規格


```

<xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
  </xsd:bizLocation>
<xsd:readpoint>22</xsd:readpoint>
<xsd:timestamp>2009-08-07T17:31:10.000+08:00</xsd:timestamp>
  </xsd:Read>
= <xsd:Read>
  <xsd:rfid>000000000000000000002221015</xsd:rfid>
  <xsd:action>OBSERVE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:shipping</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
= <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
  </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:31:10.000+08:00</xsd:timestamp>
  </xsd:Read>
  </xsd:ReadList>
  </xsd:ReadBody>
  </xsd:ReadDocument>

```

➤ 關聯事件 (範例)

關聯事件指定 ADD action 及 Packing bizStep:

```

= <xsd:ReadDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="xsd.read.sedna.com">
= <xsd:ReadBody>
= <xsd:ReadList>
= <xsd:Read>
  <xsd:parent>000000000000000000002221015</xsd:parent>
  <xsd:rfid>000000000000000000002221016</xsd:rfid>
  <xsd:action>ADD</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:packing</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
= <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
  </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>

```

```

<xsd:timestamp>2009-08-07T17:25:48.000+08:00</xsd:timestamp>
  </xsd:Read>
= <xsd:Read>
  <xsd:parent>000000000000000000002221015</xsd:parent>
  <xsd:rfid>000000000000000000002221014</xsd:rfid>
  <xsd:action>ADD</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:packing</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
= <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
  </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:25:48.000+08:00</xsd:timestamp>
  </xsd:Read>
= <xsd:Read>
  <xsd:parent>000000000000000000002221015</xsd:parent>
  <xsd:rfid>000000000000000000002221007</xsd:rfid>
  <xsd:action>ADD</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:packing</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
= <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
  </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:25:48.000+08:00</xsd:timestamp>
  </xsd:Read>
  </xsd:ReadList>
  </xsd:ReadBody>
  </xsd:ReadDocument>

```

➤ 分離事件 (範例)

分離事件指定 DELETE action 及 Disaggregate bizStep:

```
= <xsd:ReadDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="xsd.read.sedna.com">
  <xsd:ReadBody>
  <xsd:ReadList>
  <xsd:Read>
    <xsd:parent>00000000000000000000000002221016</xsd:parent>
    <xsd:rfid>00000000000000000000000002221006</xsd:rfid>
    <xsd:action>DELETE</xsd:action>
    <xsd:bizStep>urn:epcglobal:fmcg:bizstep:disaggregate</xsd:bizStep>
    <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
  <xsd:bizLocation>
    <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
    <xsd:readpoint>22</xsd:readpoint>
    <xsd:timestamp>2009-08-07T17:30:13.000+08:00</xsd:timestamp>
  </xsd:Read>
  <xsd:Read>
    <xsd:parent>00000000000000000000000002221016</xsd:parent>
    <xsd:rfid>00000000000000000000000002221002</xsd:rfid>
    <xsd:action>DELETE</xsd:action>
    <xsd:bizStep>urn:epcglobal:fmcg:bizstep:disaggregate</xsd:bizStep>
    <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
  <xsd:bizLocation>
    <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
    <xsd:readpoint>22</xsd:readpoint>
    <xsd:timestamp>2009-08-07T17:30:13.000+08:00</xsd:timestamp>
  </xsd:Read>
  <xsd:Read>
    <xsd:parent>00000000000000000000000002221016</xsd:parent>
    <xsd:rfid>00000000000000000000000002221015</xsd:rfid>
    <xsd:action>DELETE</xsd:action>
    <xsd:bizStep>urn:epcglobal:fmcg:bizstep:disaggregate</xsd:bizStep>
    <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
```

```

=> <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
  </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:30:13.000+08:00</xsd:timestamp>
  </xsd:Read>
  </xsd:ReadList>
  </xsd:ReadBody>
  </xsd:ReadDocument>

```

➤ 讀取事件 (範例)

任何吻合 XML 格式及非指定為核對、關聯、分離之事件，均會被指定為讀取事件：

```

=> <xsd:ReadDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="xsd.read.sedna.com">
=> <xsd:ReadBody>
=> <xsd:ReadList>
=> <xsd:Read>
  <xsd:rfid>000000000000000000002221006</xsd:rfid>
  <xsd:action>OBSERVE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:picking</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:in_progress</xsd:disposition>
=> <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
  </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:31:10.000+08:00</xsd:timestamp>
  </xsd:Read>
  </xsd:ReadList>
  </xsd:ReadBody>
  </xsd:ReadDocument>

```

➤ 收取已轉換 EPC 資料事件 (範例)

收取已轉換 EPC 資料事件指定由 RFID Middleware 收取之 EPC 資料格式：

```

=> <Order xmlns="com.sedna.web.epcconvert">
=> <EPCListResult>
=> <Item>

```

```
<barcode>2526451354223219884032251</barcode>  
<epc>urn:epc:id:giai:252645135.4223219884032251</epc>  
<tag>urn:epc:tag:giai-96:0.252645135.4223219884032251</tag>  
<rfid>340CF0F0F0FF00FF00FF00FB</rfid>  
</Item>
```

```
= <Item>  
<barcode>2526451354223219884032257</barcode>  
<epc>urn:epc:id:giai:252645135.4223219884032257</epc>  
<tag>urn:epc:tag:giai-96:0.252645135.4223219884032257</tag>  
<rfid>340CF0F0F0FF00FF00FF0101</rfid>  
</Item>  
</EPCListResult>  
</Order>
```

6 需求觀點

RFID Enabling Engine 是根據 EPCglobal 標準及 EPCglobal Architecture Framework 而開發，包括支援 EPC Schema 以達致與 EPC Network (如 ezTRACK) 或 RFID Middleware 之資料交換。

6.1 系統功能

RFID Enabling Engine 支援以下功能:

6.1.1 標籤打印功能

RFID Enabling Engine 支援標籤打印，包括:

- 產品層次之 RFID EPC 標籤 (SGTIN)
(根據用戶指定之 EPC Filter Value)
- 貨箱層次之 RFID EPC 標籤 (SGTIN)
(根據用戶指定之 EPC Filter Value)
- 卡板/托盤層次或貨櫃層次之 RFID EPC 標籤 (GIAI / GRAI)

根據之前 Wal*Mart 所建議之標籤格式，RFID Enabling Engine 之打印標籤功能將包括一貨箱層次之標籤樣板:

From: Sedna China Shenzhen China	To: Sam's Club USA
Ship to Postal Code: (420) 72716 	
PO #: 4890000000 LPC #: 123456789012 Case Qty: 1 SGTIN: 1234567.18901.1	
Style: MegaBox Color: Stainless Steel Silver, Mirror Finishing Size: 13.78 x 14.44 x 19.68 inch	
	
GTIN  123456789012	

6.1.2 激活標籤功能

RFID Enabling Engine 支援自動激活上傳 / 手動激活上傳功能 (EPCIS 事件)

- 自動激活上傳
如自動激活上傳已啓動，每一張標籤均會於打印時自動激活。EPCIS 事件亦會自動上傳至 EPCIS Service。

- 手動激活上傳
如自動激活上傳沒有啟動，標籤需以手動模式進行激活。當手動激活後(全部激活)，EPCIS 事件會上傳至 EPCIS Service。

6.1.3 標籤關聯功能

RFID Enabling Engine 支援以下三個層次之資料關聯:

- 產品層次標籤 與 貨箱層次標籤
- 貨箱層次標籤 與 卡板/托盤層次標籤
- 卡板/托盤層次標籤 與 貨櫃層次標籤

根據用戶於系統確認不同之關聯事件，EPCIS 事件會上傳至 EPCIS Service。

6.1.4 標籤讀取功能

RFID Enabling Engine 支援標籤讀取 (例: 由 RFID Middleware 或 RFID 硬件接收 RFID EPC 資料)及將相關 EPCIS 事件(包括讀取地點)上傳至 EPCIS Service。

6.1.5 標籤分離功能

RFID Enabling Engine 支援以下三個層次之資料分離:

- 產品層次標籤 與 貨箱層次標籤
- 貨箱層次標籤 與 卡板/托盤層次標籤
- 卡板/托盤層次標籤 與 貨櫃層次標籤

根據用戶於系統確認不同之分離事件，EPCIS 事件會上傳至 EPCIS Service。

6.1.6 標籤核對功能

RFID Enabling Engine 支援標籤核對功能，包括與系統資料進行核對。

6.1.7 手提讀取器支援功能

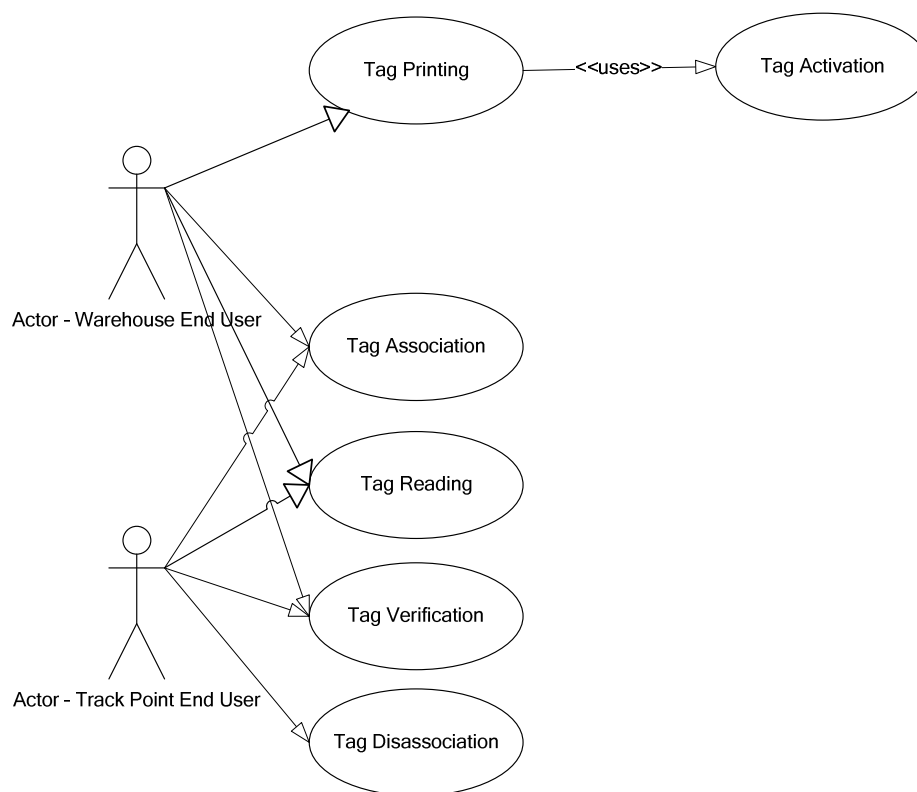
RFID Enabling Engine 支援 XML 格式之資料交換介面，包括支援接收由 RFID 手提讀取器所讀取之 RFID 資料。(XML 範例請參考本文件第五部份)

6.1.8 RFID Middleware 支援功能

RFID Enabling Engine 支援 XML 格式之資料交換介面，包括支援發送由 RFID Enabling Engine 之資料及接收由 RFID Middleware 所轉換好之 RFID EPC 資料。轉換好之 EPC 資料會於 RFID Enabling Engine 中進行下一步動作。(例: 標籤打印)

6.2 Use Cases

以下為描述 RFID Enabling Engine 中表述各功能之 Use Case Diagram:



Actor	Definition
Warehouse End User (倉庫用戶)	最初使用 RFID Enabling Engine 之用戶 (製作標籤)
Track Point End User (追蹤點用戶)	於每一物流追蹤點之 RFID Enabling Engine 用戶

6.2.1 標籤打印

6.2.1.1 描述

Use Case 描述標籤打印功能及其工作步驟

6.2.1.2 參與者

- 第一個追蹤點(貼上標籤)之倉庫員工

6.2.1.3 先設條件

- 無

6.2.1.4 主要事件流程

- 手動產品資料輸入(UPC / 產品相關資料)
- 連線至 RFID 打印機及打印標籤

6.2.1.5 替代流程

- 無

6.2.1.6 事後結果

- 衍生一樣板(單據)於系統中

6.2.2 標籤激活

6.2.2.1 描述

Use Case 描述標籤激活功能及其工作步驟

6.2.2.2 參與者

- 第一個追蹤點(貼上標籤)之倉庫員工

6.2.2.3 先設條件

- 打印標籤完成

6.2.2.4 主要事件流程

- 上傳標籤激活 EPCIS 事件
- EPCIS Service 返回接收訊息

6.2.2.5 替代流程

- 如資料上傳失敗，自動重新執行上傳
- 預設自動重新上傳間隔: 60 秒

6.2.2.6 事後結果

- 標籤激活
- 資料上傳至 EPCIS Service

6.2.3 標籤關聯

6.2.3.1 描述

Use case 描述標籤關聯功能及其工作步驟

6.2.3.2 參與者

- 第一個追蹤點之倉庫員工
- 其他追蹤點之倉庫員工

6.2.3.3 先設條件

- 無

6.2.3.4 主要事件流程

- 運用 RFID 手提讀取器讀取並關聯不同層次標籤資料
- 運用 RFID 手提讀取器傳送已關聯之資料至 RFID Enabling Engine
- 上傳標籤關聯之 EPCIS 事件資料

6.2.3.5 替代流程

- Exception thrown: 上層標籤重複讀取
- Exception thrown: EPC 未曾記錄於系統中
- 如資料上傳失敗，自動重新執行上傳
- 預設自動重新上傳間隔: 60 秒

6.2.3.6 事後結果

- 資料上傳至 EPCIS Service

6.2.4 標籤核對

6.2.4.1 描述

Use case 描述標籤核對功能及其工作步驟

6.2.4.2 參與者

- 第一個追蹤點之倉庫員工
- 其他追蹤點之倉庫員工

6.2.4.3 先設條件

- 無

6.2.4.4 主要事件流程

- 由 RFID 手提讀取器收取 RFID 標籤資料並傳送到 RFID Enabling Engine
- 核對 RFID EPC 資料完整性
- 上傳標籤核對之 EPCIS 事件資料

6.2.4.5 替代流程

- Exception thrown: 重覆標籤讀取
- Exception thrown: EPC 資料未曾在資料庫登記
- 如資料上傳失敗，自動重新執行上傳
- 預設自動重新上傳間隔: 60 秒

6.2.4.6 事後結果

- 資料上傳至 EPCIS Service

6.2.5 標籤讀取

6.2.5.1 描述

Use case 描述標籤讀取功能及其工作步驟

6.2.5.2 參與者

- 第一個追蹤點之倉庫員工
- 其他追蹤點之倉庫員工

6.2.5.3 先設條件

- 無

6.2.5.4 主要事件流程

- 實體標籤由一位置移到另一位置
- 透過 RFID 手提讀取器或 RFID Middleware 讀取標籤資料
- 登記標籤及位置資料並上傳至 EPCIS

6.2.5.5 替代流程

- Exception thrown: 重覆標籤讀取
- Exception thrown: EPC 資料未曾在資料庫登記
- 如資料上傳失敗，自動重新執行上傳
- 預設自動重新上傳間隔: 60 秒

6.2.5.6 事後結果

- 資料上傳至 EPCIS Service

6.2.6 標籤分離

6.2.6.1 描述

Use case 描述標籤分離及其工作步驟

6.2.6.2 參與者

- 第一個追蹤點之倉庫員工
- 其他追蹤點之倉庫員工

6.2.6.3 先設條件

- 標籤關聯完成

6.2.6.4 主要事件流程

- 分離不同層次之標籤資料
- 上傳標籤分離之 EPCIS 事件資料

6.2.6.5 替代流程

- Exception thrown: 重覆標籤讀取
- Exception thrown: EPC 資料未曾在資料庫登記
- 如資料上傳失敗，自動重新執行上傳
- 預設自動重新上傳間隔: 60 秒

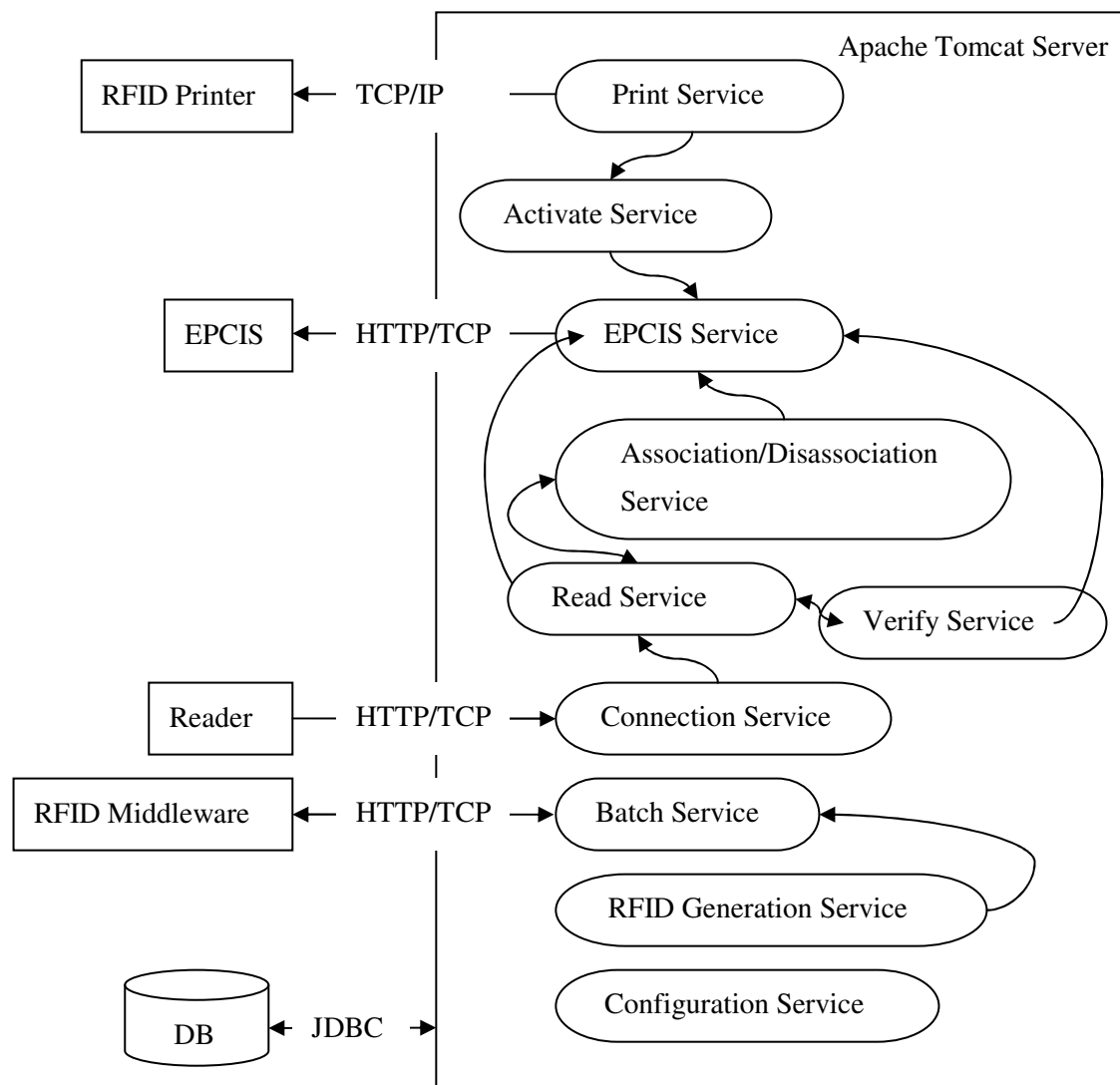
6.2.6.6 事後結果

➤ 資料上傳至 EPCIS Service

7 邏輯觀點

7.1 程序設計

RFID Enabling Engine 為一於 Apache Tomcat server 運行之 J2EE 程式並採用 JDBC 以連接資料庫。RFID Enabling Engine 同時透過 HTTP/TCP 及 TCP/IP 介面與其它外部程式進行溝通。



7.2 設計方式

RFID Enabling Engine 主要的應用邏輯均以各自的模組(Service)表達。

7.2.1 Print Service

此模組處理打印並包括以下 methods:

- Class Name
PrintDispatchAction
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	printEpcPrint(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printing(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printOrder(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
PrintDispatchActionImpl
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
byte[]	convertBytes(String printer, byte[] bytes)
String	convertChars(String printer, String string)
String	convertSGTINBarcode(String printer, String string)
String	convertSSCCBarcode(String printer, String string)
String	convertZipcode(String printer, String string)
String	generatePrintingData(String string, PrintOrder printOrder, RfidItem rfidItem, String printer)
List	printEpcPrintImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
File2String
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
String	readFileAsString(String filePath)

7.2.2 Activate Service

此模組處理激活 RFID 標籤並包括以下 methods

- Class Name
PrintDispatchAction
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
void	activateTag(List list)
TagHistory	saveActivateHistory(RfidItem rfidItem)
void	packActivateEvent(List<TagHistory> tagHistoryList)
void	packActivateEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
void	packAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	packAllActivateEvents()

7.2.3 EPCIS Service

此模組處理 EPCIS Service 並包括以下 methods:

- Class Name
schedulingUpload2GS1
- Package
org.rfidee.common.quartz
- Method List

Return Type	Declaration
void	run()

- Class Name
processState
- Package
org.rfidee.common.quartz
- Method List

Return Type	Declaration
processState	getInstance()
void	doProcess()

- Class Name
HTTPPostSender
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
URLConnection	proceed(String XML)

- Class Name
SSLUtilities
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
void	trustAllHostnames()
void	trustAllHttpsCertificates()

7.2.4 Connection Service

此模組處理與 RFID 讀取器之溝通並包括以下 methods:

- Class Name
 CallbackServerDispatchAction
- Package
 org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	SendEPC(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
 ALEHandler
- Package
 org.rfidee.module.alehandler
- Method List

Return Type	Declaration
int	process()
boolean	validate()

- Class Name
 InputHandler
- Package
 org.rfidee.module.alehandler
- Method List

Return Type	Declaration
int	process()
boolean	validate()

7.2.5 Read Service

此模組處理及分類由 RFID 讀取器傳送來之 RFID 標籤資料並包括以下 methods:

- Class Name
ReadHandler
- Package
org.rfidee.module.aehandler
- Method List

Return Type	Declaration
void	clear()
void	clearAssoDissoChildItems(String rfid, String action, String bizStep)
void	clearAssoDissoItems()
void	clearAssoDissoParentItems(String rfid, String action, String bizStep)
RfidItem	getValidDBObject(RfidItem object)
Int	handle(ECReportsDocument doc)
int	handle(ReadDocumentDocument1 doc)
boolean	isDisable()
boolean	isExists(String parentrfid, String childrfid, String bizStep, String action)
String	removeCharAt(String s, int pos)
void	setDisable(boolean disable)

- Class Name
ReadDispatchAction
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
List<ReadItem>	getReadItemList(ReadItem item)
void	packAssoEvent(List<TagHistory> tagHistoryList)
void	packAssoEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
Void	packReadEvent(List<TagHistory> tagHistoryList)
void	packVerifyEvent(List<TagHistory> tagHistoryList)
void	packVerifyEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
TagHistory	saveAssoHistory(ReadItem readItem, String parent)

TagHistory	saveVerifyHistory(ReadItem readItem)
ActionForward	clearReadItem(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	clearReadItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	readConfirmAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	readLayout(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	readRefresh(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAllRead(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

7.2.6 Verify Service

此模組處理由 RFID 讀取器傳送來之 RFID 標籤資料以進行核對並包括以下 methods:

- Class Name
VerifyHandler
- Package
org.rfidee.module.aleshandler
- Method List

Return Type	Declaration
void	clear()
boolean	isDisable()
boolean	isExists(String childrfid)
boolean	isVerifyEvent(String action, String bizStep)
void	setDisable(boolean disable)

- Class Name
VerificationDispatchAction
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
void	packVerifyEvent(List<TagHistory> tagHistoryList)
void	packVerifyEventImpl(String parent, String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
TagHistory	saveVerifyHistory(ReadItem readItem)
ActionForward	clearItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	confirmAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	removeItem(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	verificationOutput(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	verificationRefreshPage(ActionMapping mapping, ActionForm form,

	HttpServletRequest request, HttpServletResponse response)
--	---

7.2.7 Association/Disassociation Service

此模組處理由 RFID 讀取器傳送來之 RFID 標籤資料以進行關聯或分離並包括以下 methods:

- Class Name
AssoDisassoHandler
- Package
org.rfidee.module.alehandler
- Method List

Return Type	Declaration
int	AssociationCount()
boolean	checkRelationshipByRfid(String parentrfid, String childrfid)
void	clear()
int	DisAssociationCount()
boolean	isADissoAllExist(String parentrfid)
boolean	isAlreadyAsso(String rfid, String parent)
boolean	isAssoEvent(String action, String bizStep)
boolean	isDisable()
boolean	isDissoEvent(String action, String bizStep)
boolean	isExists(String parentrfid, String childrfid, String bizStep, String action)
boolean	isInvalidExists(String parentrfid, String childrfid, String bizStep, String action)
void	setDisable(boolean disable)

- Class Name
AssociationDispatchAction
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
List<ReadItem>	getReadItemList(ReadItem item)
void	packAssoEvent(List<TagHistory> tagHistoryList)
void	packAssoEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
TagHistory	saveAssoHistory(ReadItem readItem, String parent)
ActionForward	assoDisassoRefresh(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

ActionForward	clearAssoDissoItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	clearAssoDissoParentItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	confirmAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	confirmByRfid(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
int	parentSizecount(String rfid, String action, String bizStep)
ActionForward	showAllAssociation(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAllparent(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAssoDisassoLayout(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

7.2.8 Batch Service

此模組處理新增、修改及搜尋批次資料並包括以下 methods:

- Class Name
PrintDispatchAction
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	pcBack(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcDeletePattern(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcDetails(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	PCInput(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcNext(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcSample(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcSave(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printBack(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printDeletePattern(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printDetails(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printEpcList(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printEpcSearch(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printGiaiAdd(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printGraiAdd(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

ActionForward	printInput(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printNext(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printPoDelete(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printPoList(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printPoSearch(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printSample(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printSave(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printSgtinAdd(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	TagDetails(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
PrintDispatchActionImpl
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
PrintDispatchActionImpl	getInstance()
void	clearTemplateGaiImpl(ActionForm form)
void	clearTemplateGrailImpl(ActionForm form)
void	clearTemplateSgtinImpl(ActionForm form)
List	pcPatternListImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, boolean flag)
void	pcSampleImpl(ActionForm form)
PrintOrder	pcSaveImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
PrintOrder	printDetailsImpl(ActionMapping mapping, ActionForm form,

	HttpServletRequest request, HttpServletResponse response)
RfidItem	printEpcSearchImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printGiaiCleanImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Map	printGiaiVarInputImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Pattern	printGiaiVarOutputImpl(Map<String, String> map)
void	printGraiCleanImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Map	printGraiVarInputImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Pattern	printGraiVarOutputImpl(Map<String, String> map)
void	printNextImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
List	printOrderImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
List	printPatternListImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, boolean flag)
PrintOrder	printPoDeleteImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
PrintOrder	printPoSearchImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printSampleImpl(ActionForm form)
PrintOrder	printSaveImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printSgtinCleanImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Map	printSgtinVarInputImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Pattern	printSgtinVarOutputImpl(Map<String, String> map)
void	TagDetailsImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, RfidItem rfidItem)

- Class Name
CheckGiai
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
List	check()
ActionErrors	getErrorMessages()

- Class Name
CheckGrai
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
List	check()
ActionErrors	getErrorMessages()

- Class Name
CheckSgtin
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
List	check()
ActionErrors	getErrorMessages()

- Class Name
InsertZero
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
String	insert(String s, int len)

7.2.9 RFID Generation Service

此模組處理及產生 RFID 標籤順序號並包括以下 methods:

- Class Name
PrintDispatchActionImpl
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
String	getPrefix(String str)
Boolean	GenerateRfidImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, PrintOrder printOrder)
Boolean	GenerateRfidWithOutDataplexImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, PrintOrder printOrder)
int	GraiGetCheckDigit(String PreGRAI)
String	HexDecConversion(String Hex, int Dec, boolean hextoDec)
String	increment(String str, int length, int value)
String	incrementHex(String str, int length, int value)
void	saveRfidItem(String epc, PrintOrder printOrder, Pattern pattern)
String	sendtoDataplex(String url,String content)
int	SgtinGetCheckDigit(String GTIN13)

- Class Name
HTTPPostSender
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
URLConnection	proceed(String XML)

- Class Name
SSLUtilities
- Package
org.rfidee.common.util
- Method List

Return Type	Declaration
void	trustAllHostnames()
void	trustAllHttpsCertificates()

7.2.10 Configuration Service

此模組處理系統設定並包括以下 methods:

- Class Name
AdministrationDispatchAction
- Package
org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	adminConfig(ActionMapping mapping
ActionForward	adminConfigError(ActionMapping mapping
ActionForward	adminSave(ActionMapping mapping

8 進程觀點

8.1 RFID Enabling Engine - RFID 硬件溝通介面

RFID Enabling Engine 透過呼叫 RFID 打印機 API 以支援 RFID 標籤打印，並以 TCP / IP 模式與 RFID 打印機進行溝通。

RFID Enabling Engine 支援 XML 檔案格式並透過 HTTP 模式與 RFID 讀取器進行溝通。CallbackServerDispatchAction (*java interface class*)負責收集 RFID 資料。同時，於 org.rfidee.module.alexhandler 中其他 java classes 則負責處理該接收資料。

8.2 RFID Enabling Engine – EPCIS 溝通介面

RFID Enabling Engine 預設了不同 EPCIS 事件於系統中，包括激活事件、核對事件、關聯事件及分離事件。當系統接收到相關事件資料後，會根據情況轉換為 EPCIS 事件資料。之後，schedulingUpload2GS1(Scheduler Class)於系統中透過 HTTP 連接模式傳送相關事件資料至 EPCIS。同時，schedulingUpload2GS1 會監控著上傳狀態並於上傳失敗時自動進行再上傳。(XML 範例請參閱第五部份)

8.3 RFID Enabling Engine - RFID Middleware 溝通介面

RFID Enabling Engine 定義了相關 XML 格式與 RFID Middleware 進行溝通。當系統產生了相應之 SGTIN、GIAI 或 GRAI 順序號後，系統可將該資料傳送至 RFID Middleware 以進行 EPC 資料轉換。轉換好之 EPC 資料會傳送回 RFID Enabling Engine 以進行後續工作(如: 打印標籤)。

以下為系統傳送資料到 RFID Middleware 之 XML 範例:

```
= <Order xmlns="com.sedna.web.epcconvert">
= <EPCList>
= <GiaiItem>
  <company_prefix>252645135</company_prefix>
  <serial_reference>4223219884032251</serial_reference>
  <serial_end_reference>4223219884032257</serial_end_reference>
  <epc_filter>0</epc_filter>
  </GiaiItem>
</EPCList>
</Order>
```

以下為 RFID Middleware 傳送資料回系統之 XML 範例:

```
= <Order xmlns="com.sedna.web.epcconvert">
= <EPCListResult>
= <Item>
  <barcode>2526451354223219884032251</barcode>
  <epc>urn:epc:id:giai:252645135.4223219884032251</epc>
  <tag>urn:epc:tag:giai-96:0.252645135.4223219884032251</tag>
  <rfid>340CF0F0F0FF00FF00FF00FB</rfid>
  </Item>
= <Item>
  <barcode>2526451354223219884032257</barcode>
  <epc>urn:epc:id:giai:252645135.4223219884032257</epc>
  <tag>urn:epc:tag:giai-96:0.252645135.4223219884032257</tag>
  <rfid>340CF0F0F0FF00FF00FF0101</rfid>
  </Item>
  </EPCListResult>
</Order>
```

9 實施觀點

9.1 資料庫及程式伺服器考量

9.1.1 MySQL

MySQL 是世上最流行之開放源代碼資料庫。其優點在於處理速度快、高穩定性、高可用性及擁有充足技術支援。同時，MySQL 可支援不同操作平台，包括: Window、Linux、HP-UX、AIX 等。於系統程式編寫上，MySQL 亦支援一般常用之資料庫 API 如 ODBC 及 JDBC 等，給系統開發者提供了不少方便。另外，MySQL 作為一合適之應用資料庫，可支援超過 7TB 之數據量及過百萬列資料而不影響系統運作。

9.1.2 Apache Tomcat

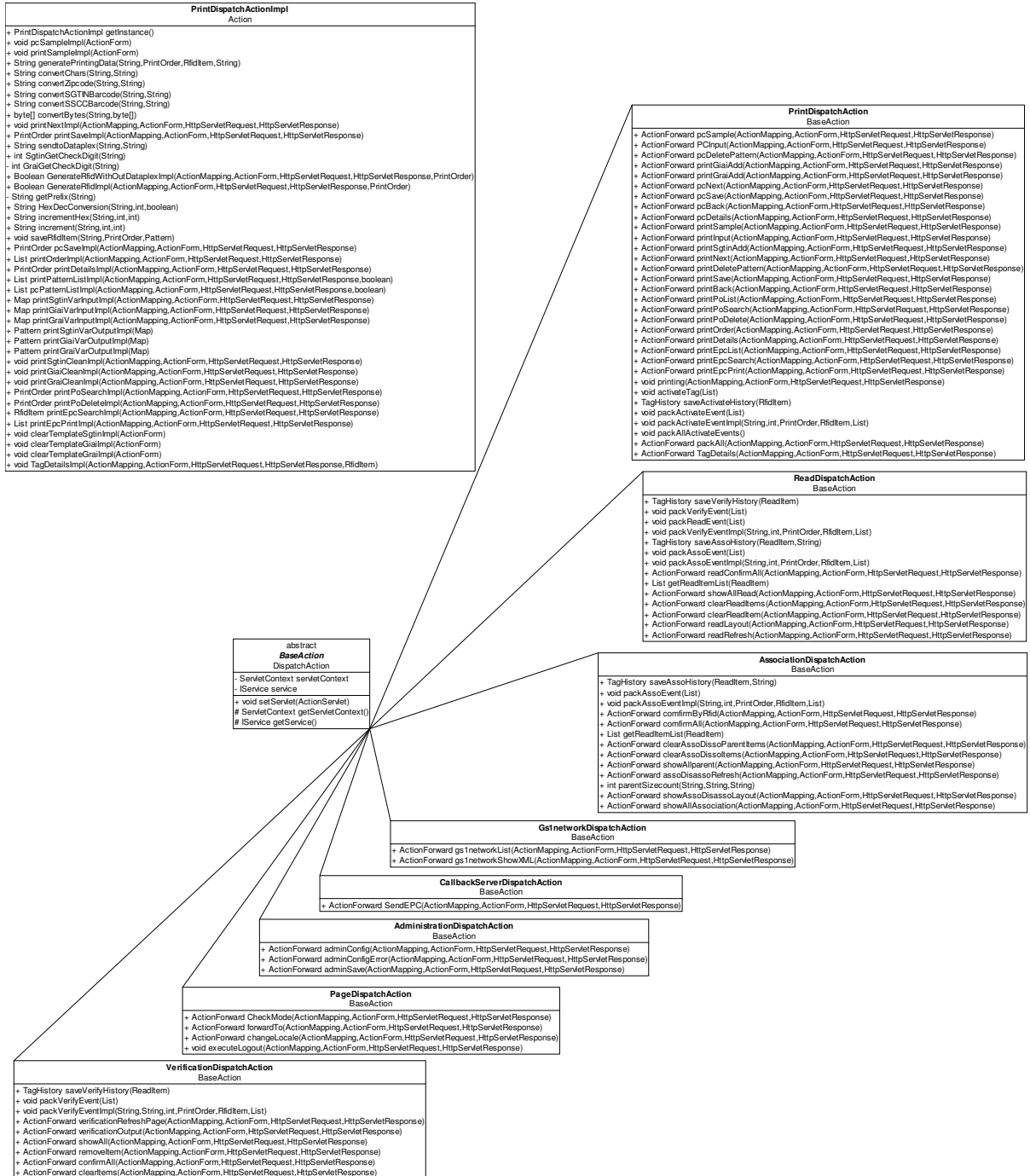
Apache Tomcat 是由 Apache 軟件基金會開發的一個 Servlet 容器。按照 Sun Microsystems 提供的技術規範，實現了對 Servlet 和 JavaServer Page (JSP) 的支持，並提供了作為 Web 服務器的一些特有功能，如 Tomcat 管理和控制平台、安全域管理等。同時，用戶可簡單地透過修改 XML 設定檔以修改 Tomcat 設定。

9.2 Class Diagram

以下為每一個 Package 下之 Class Diagram 以表達相對應之 methods 及 properties。

9.2.1 org.rfidee.web

9.2.1.1 action



9.2.2.2 quartz

schedulingUpload2GS1 Object
+ void run()

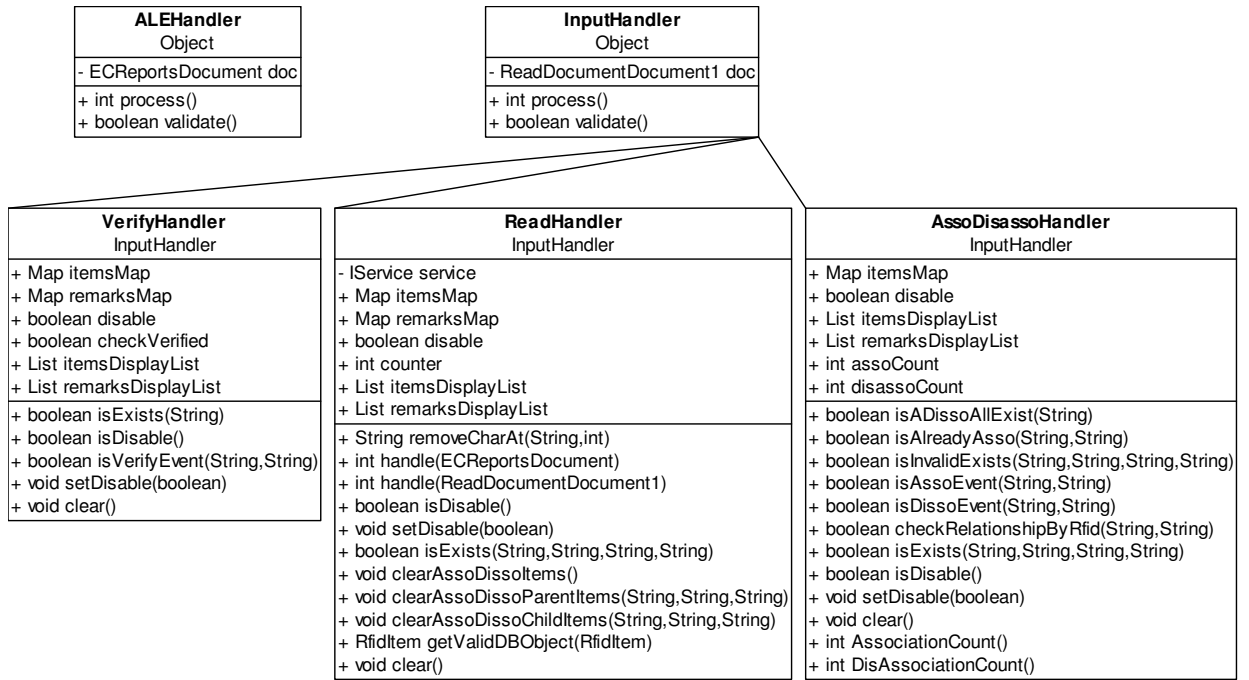
processState Object
+ processState instance
+ processState getInstance()
+ void doProcess()

9.2.2.3 util

AdministrationDisplayDecorator TableDecorator	AssociationDisplayDecorator TableDecorator	CheckGlaI Object	CheckGraI Object	CheckSgtn Object	DateUtil Object	File2String Object	GstnetworkDisplayDecorator TableDecorator
	+ int count + String getTime() + String getDate() + String getEvent() + String getRemovebutton()	- Map map - List patternList - List tempList - ActionErrors errors - List check() + ActionErrors getErrorMessages()	- Map map - List patternList - List tempList - ActionErrors errors - List check() + ActionErrors getErrorMessages()	- Map map - List patternList - List tempList - ActionErrors errors - List check() + ActionErrors getErrorMessages()	+ String DB_DATETIME_FORMAT + String JS_DATETIME_FORMAT + String DATE_FORMAT + String EXPORT_DATETIME_FORMAT - SimpleDateFormat df + Calendar getCalendar() + Calendar toCalendar(String) + Calendar toCalendar(String,String) + Calendar toCalendar(int,int,int) + String format(Calendar,String) + Calendar getCurrentMonthFirstDay() + Calendar getCurrentMonthLastDay() + int getMonthLastDay(int) + Calendar getNextWeek(Calendar) + Calendar getNextMonth(Calendar) + Calendar getNextDay(Calendar) + String getDateIntervalByWeek(int,int,String)	+ String readFileAsString(String)	- DateFormat dateFormat + String getSubmission_status() + String getSubmission_date()
HTTPPostSender Object	InserZero Object	PrintDisplayDecorator TableDecorator	SpringUtil Object	SSLUtilifies Object	VerificationDisplayDecorator TableDecorator		
+ int serial + String filename + String encoding + String url + boolean https + boolean auth + String username + String password + boolean isAuth() + void setAuth(boolean) + String getEncoding() + void setEncoding(String) + String getFilename() + void setFilename(String) + boolean isHttps() + void setHttps(boolean) + String getPassword() + void setPassword(String) + String getURL() + void setURL(String) + String getUsername() + void setUsername(String) + URLConnection proceed(String) + URLConnection proceedWithFile()	+ String insert(String,int)	- DateFormat dateFormat + String getTime() + String getDate() + String getEvent() + String getCreation_date() + String getPattern_content() + String getPattern_content2() + String getLabelType() + String getPostfix(String) + String getPatternURL() + String getEponferenceno() + String getPreferenceno()	+ Object getBean(String)	- HostnameVerifier _hostnameVerifier - TrustManager _trustManagers[] - HostnameVerifier _hostnameVerifier - TrustManager _trustManagers[] - void _trustAllHostnames() - void _trustAllHttpsCertificates() - boolean isDeprecatedSSLProtocol() - void _trustAllHostnames() - void _trustAllHttpsCertificates() + void trustAllHostnames() + void trustAllHttpsCertificates()			

9.2.3 org.rfidee.module

9.2.3.1 alehandler



9.2.3.2 dao

9.2.3.2.1 iface

abstract IDAIO Object
+ void saveRecord(Object) + void saveListRecord(List) + void deleteRecord(Object) + void deleteListRecord(List) + PrintOrder findItemById(PrintOrder) + PrintOrder findItemByIdNoRfidNoPattern(PrintOrder) + PrintOrder findItemByIdNoRfid(PrintOrder) + List findAllItem(PrintOrder) + List searchPrint(PrintOrder) + List findChildRfidByParentId(RfidItem) + RfidItem findRfidById(RfidItem) + List findAllRfid(RfidItem) + List searchRfid(RfidItem) + int findNumberOfValidRfidByPrintorder(PrintOrder) + EpcisEvent findEventBySpecname(EpcisEvent) + EpcisDatastore findDataById(EpcisDatastore) + List findAllData() + List findAllDataByUpdated(EpcisDatastore) + Config findConfig() + void updateSuccessCiRfidItem(long) + void updateSuccessCiTagHistory(long) + List findAllUnpackedActivateEvents(TagHistory) + List findTagHistoryByEPC(TagHistory) + int findNumberOfParentInDB(long) + List findAllChildWithParentID(long) + RfidItem findRfidItemByRfidObject(RfidItem)

9.2.3.2.2 impl

implements IDAIO MysqlHibernateDAO HibernateDaoSupport
+ void saveRecord(Object) + void saveListRecord(List) + void deleteRecord(Object) + void deleteListRecord(List) + PrintOrder findItemById(PrintOrder) + PrintOrder findItemByIdNoRfidNoPattern(PrintOrder) + PrintOrder findItemByIdNoRfid(PrintOrder) + List findAllItem(PrintOrder) + List searchPrint(PrintOrder) + List findChildRfidByParentId(RfidItem) + RfidItem findRfidById(RfidItem) + PrintOrder findItemByIdWithoutList(PrintOrder) + int findNumberOfValidRfidByPrintorder(PrintOrder) + List findAllRfid(RfidItem) + List searchRfid(RfidItem) + EpcisEvent findEventBySpecname(EpcisEvent) + EpcisDatastore findDataById(EpcisDatastore) + List findAllData() + List findAllDataByUpdated(EpcisDatastore) + Config findConfig() + void updateSuccessCiRfidItem(long) + void updateSuccessCiTagHistory(long) + List findAllUnpackedActivateEvents(TagHistory) + List findTagHistoryByEPC(TagHistory) + int findNumberOfParentInDB(long) + List findAllChildWithParentID(long) + RfidItem findRfidItemByRfidObject(RfidItem)

9.2.3.3 service

9.2.3.3.1 iface

abstract IService Object
+ void saveRecord(Object) + void saveListRecord(List) + void deleteRecord(Object) + void deleteListRecord(List) + PrintOrder findItemByld(PrintOrder) + PrintOrder findItemByldNoRfidNoPattern(PrintOrder) + PrintOrder findItemByldNoRfid(PrintOrder) + List findAllItem(PrintOrder) + List searchPrint(PrintOrder) + List findChildRfidByParentId(RfidItem) + RfidItem findRfidByld(RfidItem) + List findAllRfid(RfidItem) + List searchRfid(RfidItem) + EpcisEvent findEventBySpecname(EpcisEvent) + EpcisDatastore findDataByld(EpcisDatastore) + List findAllData() + List findAllDataByUpdated(EpcisDatastore) + Config findConfig() + void updateSuccessCl(EpcisDatastore) + List findAllUnpackedActivateEvents(TagHistory) + List findTagHistoryByEPC(TagHistory) + int findNumberOfParentInDB(long) + List findAllChildWithParentID(long) + RfidItem findRfidItemByRfidObject(RfidItem) + int findNumberOfValidRfidByPrintorder(PrintOrder)

9.2.3.3.2 impl

implements IService
ServiceImpl
Object

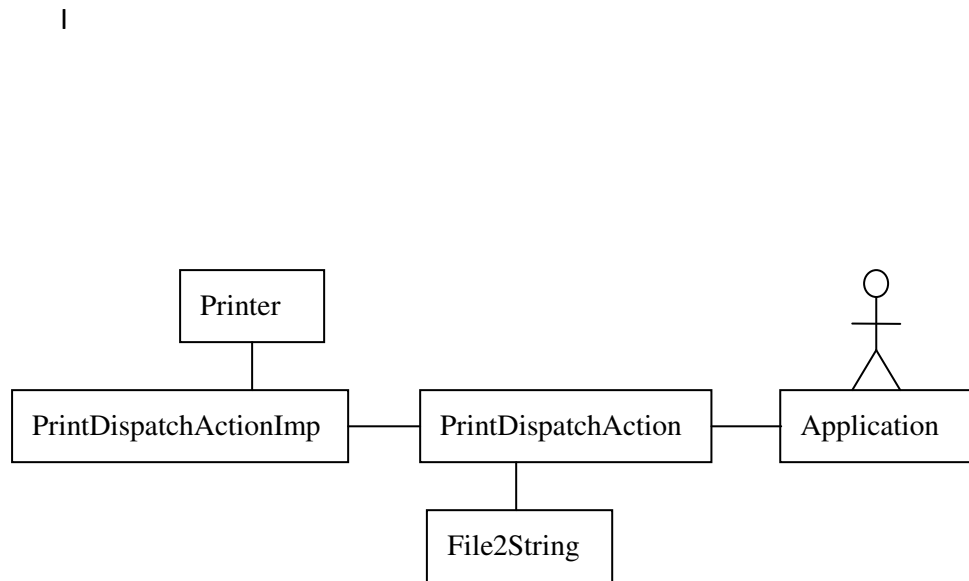
- IDAO dao
+ IDAO getDao() + void setDao(IDAO) + void saveRecord(Object) + void saveListRecord(List) + void deleteListRecord(List) + void deleteRecord(Object) + PrintOrder findItemByld(PrintOrder) + PrintOrder findItemByldNoRfidNoPattern(PrintOrder) + PrintOrder findItemByldNoRfid(PrintOrder) + List findAllItem(PrintOrder) + List searchPrint(PrintOrder) + List findChildRfidByParentId(RfidItem) + RfidItem findRfidByld(RfidItem) + List findAllRfid(RfidItem) + List searchRfid(RfidItem) + EpcisEvent findEventBySpecname(EpcisEvent) + EpcisDatastore findDataByld(EpcisDatastore) + List findAllData() + List findAllDataByUpdated(EpcisDatastore) + Config findConfig() + void updateSuccessCl(EpcisDatastore) + List findAllUnpackedActivateEvents(TagHistory) + List findTagHistoryByEPC(TagHistory) + int findNumberOfParentInDB(long) + List findAllChildWithParentID(long) + RfidItem findRfidItemByRfidObject(RfidItem) + int findNumberOfValidRfidByPrintorder(PrintOrder)

9.3 Collaboration Diagram

以下為各主要 Services 之 Collaboration Diagram。每一個工序裡之主要 Entities 將會描繪於圖中。

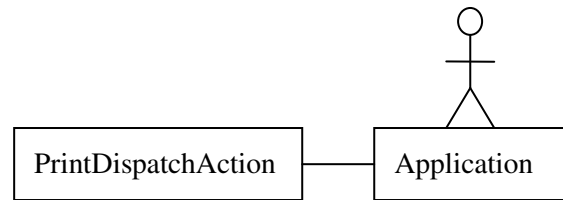
9.3.1 Print Service

於 Print Servicez 中，打印工序主要由 Class PrintDispatchAction 處理。File2String 及 PrintDispatchActionImp 均為支援 PrintDispatchAction。當資料處理完畢，PrintDispatchAction 會傳送所需打印資料至打印機以打印相關標籤。



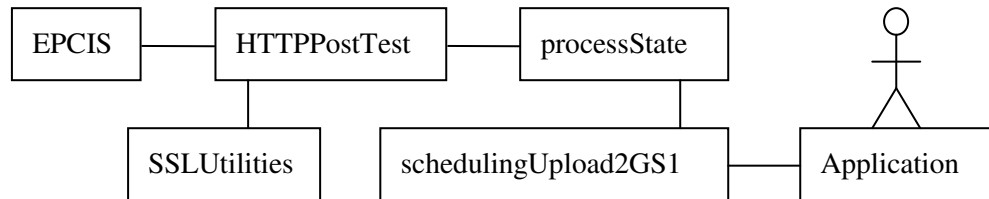
9.3.2 Activate Service

系統內部運作如下圖：



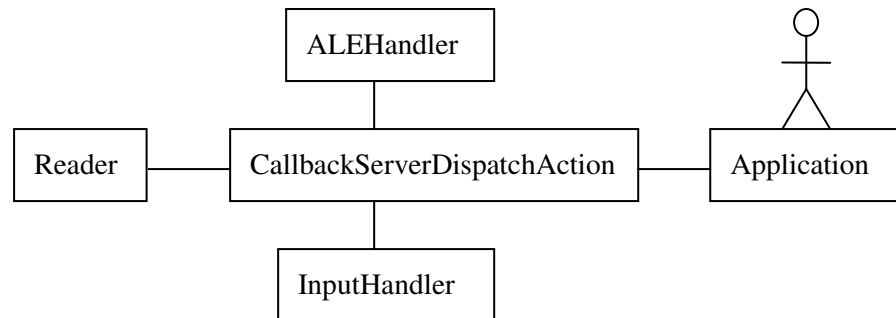
9.3.3 EPCIS Service

於 EPCIS Service 中，Class schedulingUpload2GS1 會重覆觸發 processState 並要求 processState 每數分鐘檢視系統資料一次以判斷有否系統資料需要上傳至 EPCIS。如有資料需上傳至 EPCIS，processState 會透過 HTTPPostTest 及 SSLUtilities 上傳資料至 EPCIS。



9.3.4 Connection Service

於 Connection Service 中，Class CallbackServerDispatchAction 主要用作連接讀取器及接收相關資料。之後，CallbackServerDispatchAction 會在進行下一步動作前分類及調配資料至 ALEHandler 及 InputHandler 以作資料核對之用。



9.3.5 Read Service

於 Read Service 中，Class ReadDispatchAction 負責分類、處理及顯示由讀取器接收之資料。同時 ReadDispatchAction 亦負責包裝好資料以上傳至 EPCIS。所有 ReadDispatchAction 下之 action 均透過 ReadHandler 觸發。



9.3.6 Verify Service

於 Verify Service 中，Class VerificationDispatchAction 負責分類、處理及顯示由讀取器接收之資料。同時 VerificationDispatchAction 亦負責包裝好資料以上傳至 EPCIS。所有 VerificationDispatchAction 下之 action 均透過 VerifyHandler 觸發。



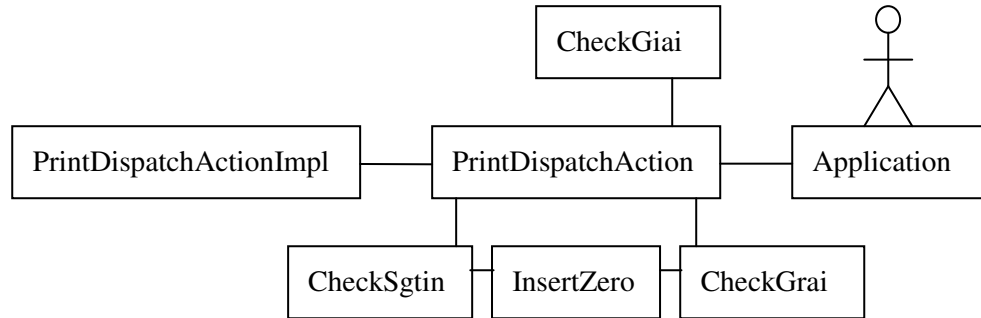
9.3.7 Association/Disassociation Service

於 Association/Disassociation Service 中，Class AssociationDispatchAction 負責分類、處理及顯示由讀取器接收之資料。同時 AssociationDispatchAction 亦負責包裝好資料以上傳至 EPCIS。所有 AssociationDispatchAction 下之 action 均透過 AssoDisassoHandler 觸發。



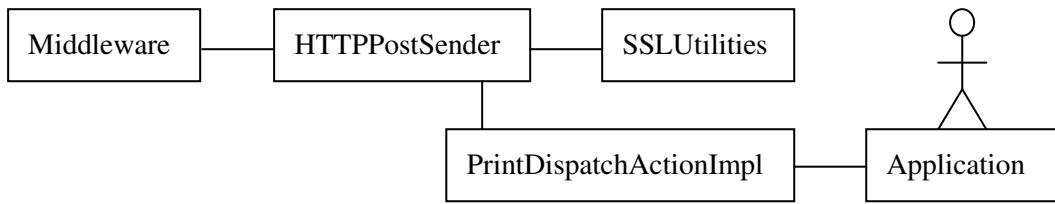
9.3.8 Batch Service

於 Batch Service 中，Class PrintDispatchAction 負責處理批次(樣板)。所有 PrintDispatchAction 下之 action 均透過 PrintDispatchActionImpl、CheckGiai、CheckSgtin、CheckGrai 及 InsertZero 觸發。



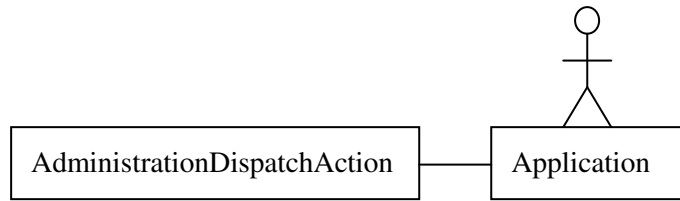
9.3.9 RFID Generation Service

於 RFID Generation Service 中，Class PrintDispatchActionImpl 負責處理產生 RFID 標籤順序號。另一方面，PrintDispatchActionImpl 亦可根據系統設定連接上 RFID Middleware 以產生 RFID 標籤順序號。該產生工作會傳送至 RFID Middleware 並透過 HTTPPostSender 及 SSLUtilities 以接收 RFID 標籤資料。



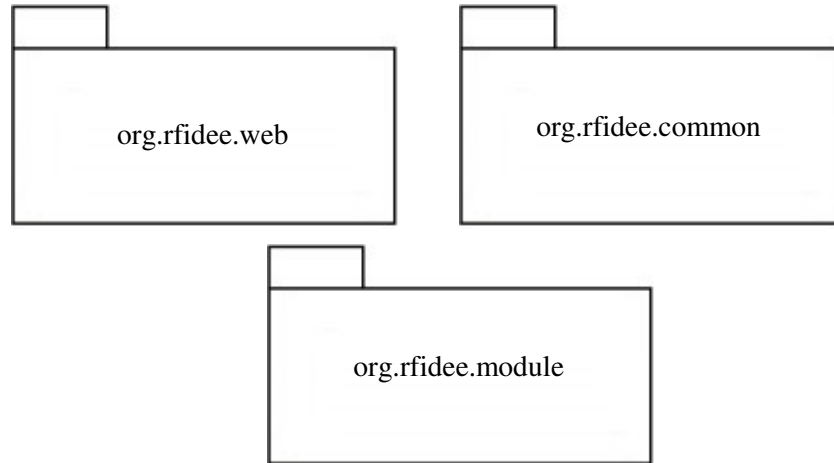
9.3.10 Configuration Service

系統內部運作如下圖：



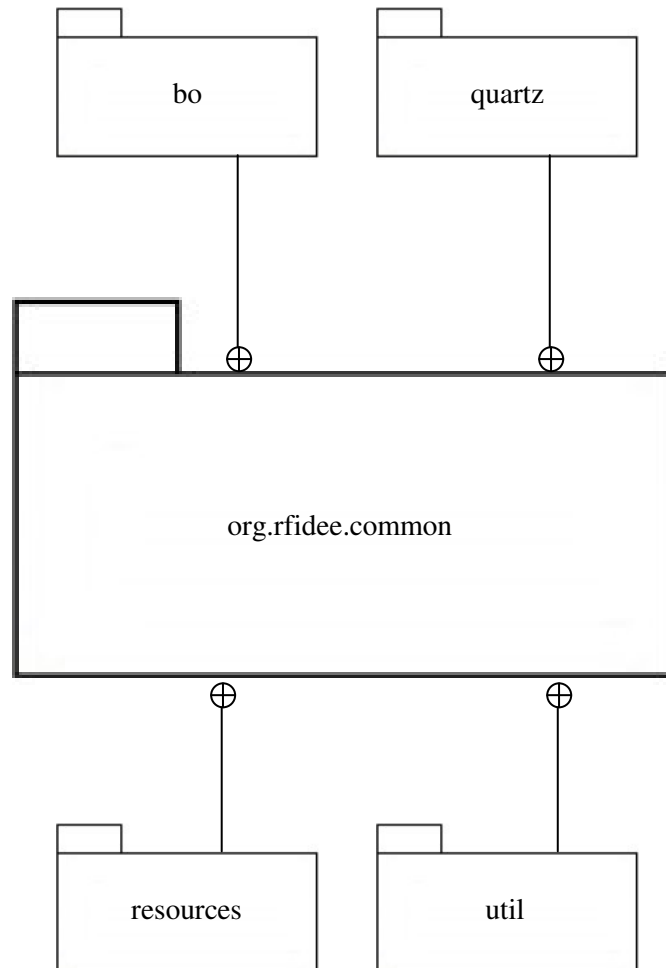
9.4 Package Diagram

RFID Enabling Engine 主要由 3 個 Java packages 組成:
org.rfidee.common, org.rfidee.module and org.rfidee.web.



9.4.1 org.rfidee.common

“org.rfidee.common” package 包含主要構成 RFID Enabling Engine 內部 Classes:

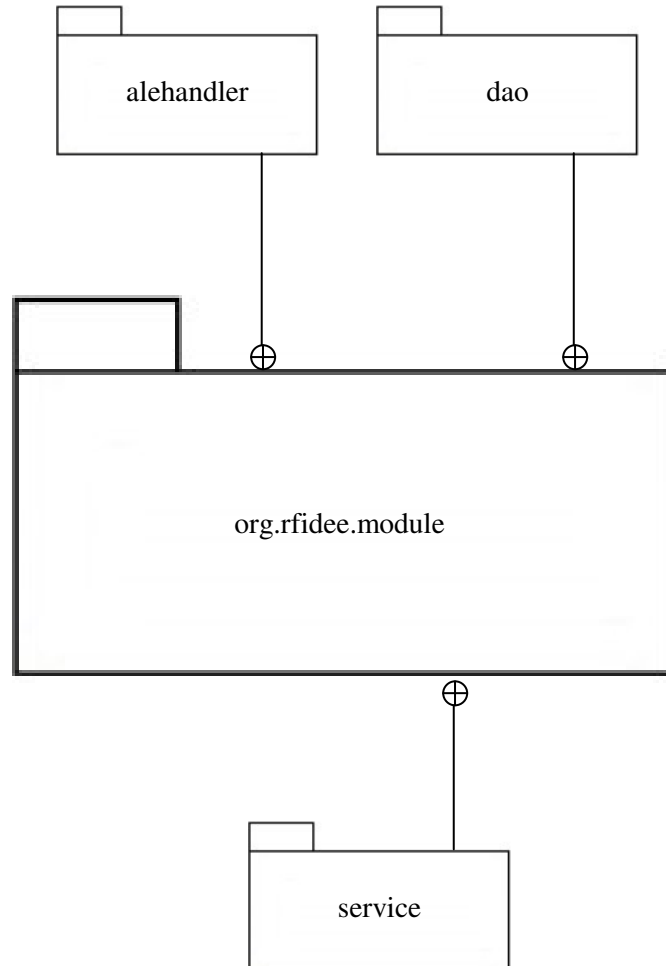


於 org.rfidee.common 下之 Package:

- bo: 包含 Data Access Objects 以負責資料庫存取。
- quartz: 包含所有支援主要實施用 Class 之排程用 Class
- resources: 包含多類型語言檔案以供顯示介面之用，並無包含任何 Class。
- util: 包含 Class 以支援主要 Class 運行之用

9.4.2 org.rfidee.module

“org.rfidee.module” package 包含主要構成 RFID Enabling Engine 內部之餘下 Classes:

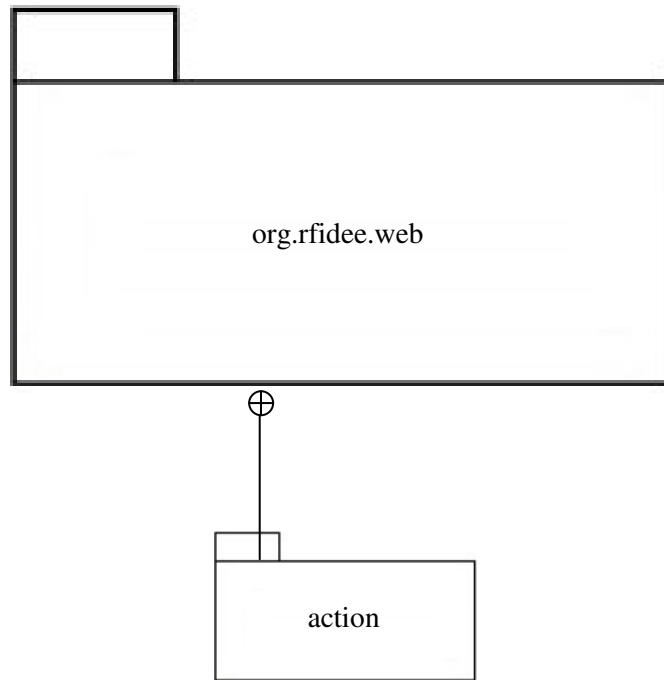


於 org.rfidee.module 下之 Package:

- alehandler: 包含支援 Class 以支援主要構成 Class
- dao: 包含主要構成 Class 並與資料庫資料進行互動
- service: 包含主要構成 Class 以運行主要功能

9.4.3 org.rfidee.web

“org.rfidee.web” package 包含用戶網上介面及其互動功能 Classes:



於 `org.rfidee.web` 下之 Package:

- `action`: 包含用戶網上介面及其互動功能

10 部署觀點

10.1 運作環境

10.1.1 設置

RFID Enabling Engine 標準運作環境包括:

- Apache Tomcat 5.5
- Java Development Kit (JDK) 1.5
- MySQL 4.1 +MySQLGUI 5.0

10.1.2 建立及部署

- 使用 Apache Ant 1.7.1 建立項目
- 將 war 檔案放置於 Apache Tomcat 5.5 home 路徑下之 webapps 資料夾中

11 資料觀點

11.1 數據庫表 config

此數據庫表儲存系統設定資料。

Column name	Type	Description
Id	BIGINT(20)	Primary key
serverIP	VARCHAR(255)	The server IP of middleware
printerIP	VARCHAR(255)	Printer setting
printerPort	VARCHAR(255)	Printer setting
printerFilePath	VARCHAR(255)	Printer setting
epcisServerUrl	VARCHAR(255)	EPCIS setting
epcisServerLogin	VARCHAR(255)	EPCIS setting
epcisServerPassword	VARCHAR(255)	EPCIS setting
autoActivate	TINYINT(1)	Activate setting
printer	VARCHAR(255)	Printer setting

Primary Key: id

11.2 數據庫表 epics_datastore

此數據庫表儲存需要上傳至 EPCIS 之資料。

Column name	Type	Description
ci_id	BIGINT(20)	Primary key
epcis_event_id	BIGINT(20)	Foreign key – epcis_event
xmlcontent	LONGTEXT	The upload data to EPCIS in XML format
creation_date	DATETIME	The date and time of creation
submission_date	DATETIME	The date and time of submission to EPC network
updated	TINYINT(1)	Submission status
count	BIGINT(20)	Number of EPC elements
printorderno	VARCHAR(255)	Batch No related
printorderid	BIGINT(20)	Foreign key – printorder
submit_url	VARCHAR(255)	Submission destination
submit_login	VARCHAR(255)	Login name to EPC Network
return_code	VARCHAR(255)	HTTP Code of EPC server return

Primary Key: ci_id

11.3 數據庫表 epcis_event

此數據庫表儲存 EPCIS 事件資料。

Column name	Type	Description
eventID	BIGINT(20)	Primary key
specname	VARCHAR(255)	The internal system key, spec name of the event
bizstepname	VARCHAR(255)	The EPCIS key, business step name
action	VARCHAR(255)	The EPCIS key, action of the event
bizlocation	VARCHAR(255)	The EPCIS key, business location
readpoint	VARCHAR(255)	The EPCIS key, read point
dispositionId	VARCHAR(255)	The EPCIS key, disposition ID
template	LONGTEXT	Upload template file

Primary Key: eventID

11.4 數據庫表 pattern

此數據庫表儲存資料以供系統內部運作。

Column name	Type	Description
id	BIGINT(20)	Primary key
pattern_company_prefix	VARCHAR(255)	Company prefix for generating RFID
pattern_sgtin_indicator	VARCHAR(255)	SGTIN indicator for generating RFID
pattern_sgtin_item_reference	VARCHAR(255)	SGTIN item reference for generating RFID
pattern_sgtin_epc_filter	VARCHAR(255)	EPC filter for generating RFID
pattern_serial_reference	VARCHAR(255)	Serial for generating RFID
pattern_end_serial	VARCHAR(255)	The end of serial for generating RFID
pattern_type	VARCHAR(255)	RFID generation type, SGTIN, GIAI or GRAI
printorder_id	BIGINT(20)	Foreign key – printorder
pattern_sgtin_label_type	VARCHAR(255)	The type of SGTIN, Carton
pattern_upc	VARCHAR(255)	UPC information
pattern_case_qty	VARCHAR(255)	General information field
pattern_style	VARCHAR(255)	General information field
pattern_color	VARCHAR(255)	General information field
pattern_size	VARCHAR(255)	General information field
pattern_string	VARCHAR(255)	Generation result
pattern_giai_label_type	VARCHAR(255)	The type of GIAI, Pallet
pattern_grai_label_type	VARCHAR(255)	The type of GRAI, Container
pattern_grai_asset_type	VARCHAR(255)	The asset type of GRAI

Primary Key: id

11.5 數據庫表 printorder

此數據庫表儲存每一張標籤之輸入資料。

Column name	Type	Description
id	BIGINT(20)	Primary key
s_purchase_order_no	VARCHAR(255)	Batch number
s_factory_name	VARCHAR(255)	Sender address
s_address	VARCHAR(255)	Sender address
s_city	VARCHAR(255)	Sender address
s_state_region	VARCHAR(255)	Sender address
s_zip_postal_code	VARCHAR(255)	Sender address
s_country	VARCHAR(255)	Sender address
r_factory_name	VARCHAR(255)	Recipient address
r_address	VARCHAR(255)	Recipient address
r_city	VARCHAR(255)	Recipient address
r_state_region	VARCHAR(255)	Recipient address
r_zip_postal_code	VARCHAR(255)	Recipient address
r_country	VARCHAR(255)	Recipient address
r_ship_to_postal_code	VARCHAR(255)	Recipient address
creation_date	DATETIME	The creation time
total	VARCHAR(255)	The total number of tags
s_factory_name2	VARCHAR(255)	Sender address
r_factory_name2	VARCHAR(255)	Recipient address
s_factory_name_s	VARCHAR(255)	Sender address
r_factory_name_s	VARCHAR(255)	Recipient address
order_type	VARCHAR(255)	The internal system key

Primary Key: id

11.6 數據庫表 rfiditem

此數據庫表儲存每一張標籤之所有資料。

Column name	Type	Description
id	BIGINT(20)	Primary key
parent	BIGINT(20)	The internal system ID for association and Disassociation
epc	VARCHAR(255)	The EPC number of a tag
tag	VARCHAR(255)	The Tag number of a tag
rfid	VARCHAR(255)	The RFID hex number of a tag
application_identifier	VARCHAR(255)	Application identifier for GIAI and GRAI
barcode	VARCHAR(255)	The barcode of a tag
type	VARCHAR(255)	The type of EPC number
print_status	TINYINT(1)	//0=not printed; 1=printed
print_date	DATETIME	The date and time of printing
printorder_id	BIGINT(20)	Foreign key – printorder
upc	VARCHAR(255)	The UPC of a tag
case_qty	VARCHAR(255)	Tag information
style	VARCHAR(255)	Tag information
size	VARCHAR(255)	Tag information
color	VARCHAR(255)	Tag information
label_type	VARCHAR(255)	The level type //Item = item level; Carton = carton level; Pallet = pallet level; Container = container level
ci_id	BIGINT(20)	Foreign key – epcis_datastore
ci_date	DATETIME	Date of the foreign key
status	VARCHAR(255)	The internal system key
pattern_id	VARCHAR(255)	Foreign key – pattern
xml_id	BIGINT(20)	Reserved field
order_type	VARCHAR(255)	The internal system key
activate_status	TINYINT(1)	//0=not activated; 1=activated
activate_date	DATETIME	Date of activation

Primary Key: id

11.7 數據庫表 taghistory

此數據庫表儲存由讀取器傳送至系統之資料。

Column name	Type	Description
id	BIGINT(20)	Primary key
parent	VARCHAR(255)	The internal system ID for association and Disassociation
epc	VARCHAR(255)	Data from reader, EPC
action	VARCHAR(255)	Data from reader, action of the event
bizStep	VARCHAR(255)	Data from reader, business step name
disposition	VARCHAR(255)	Data from reader, disposition ID
bizLocation	VARCHAR(255)	Data from reader, business location
readpoint	VARCHAR(255)	Data from reader, read point
timestamp	TIMESTAMP	Timestamp of record
ci_id	BIGINT(20)	Foreign key – epcis_datastore
ci_date	DATETIME	Date of the foreign key
printorder_id	BIGINT(20)	Foreign key – printorder
rfiditem_id	BIGINT(20)	Foreign key – rfiditem

Primary Key: id

12 系統特性

RFID Enabling Engine 根據 J2EE 技術開發並延伸其一貫標準特性。

12.1 延伸性

RFID Enabling Engine 根據 JAVA 物件導向結構及其嚴謹 Class Library 架構開發。開發者可以 Sub-classes 形式延伸已開發之系統功能從而新增其他自定義之系統功能。

12.2 穩定性

根據 JAVA 其久經測試之規格及架構，RFID Enabling Engine 可確保其編寫程式及資料之穩定性及完整性。詳細之硬件要求及系統運作環境設置會清晰列於指引中以保證用戶正常運作 RFID Enabling Engine。

12.3 多樣性

RFID Enabling Engine 不僅支援與 EPCIS 之溝通介面，並同時支援直接(API procedure call)及間接(透過 RFID Middleware)與 RFID 硬件之溝通介面。

12.4 可攜性

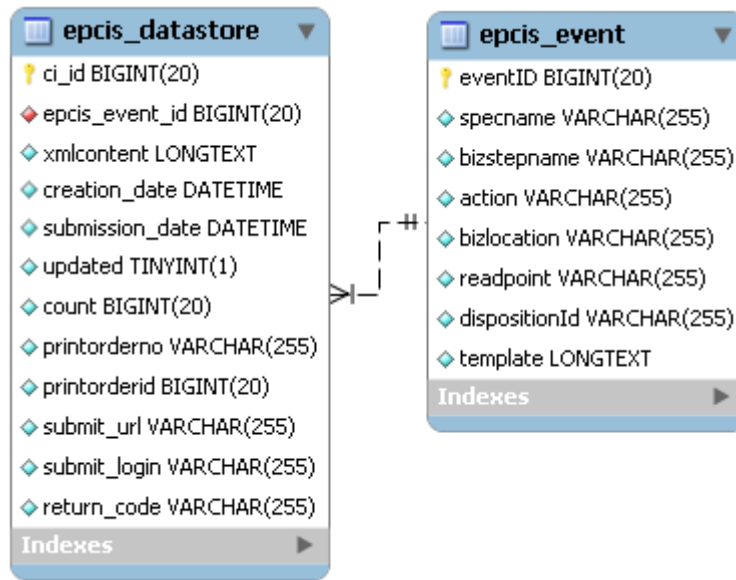
由於 JAVA 網絡程式為非依賴特定操作系統之程式，其程式可運行於不同操作系統介面包括 Win XP 及 Linux。同時，作為網絡程式，RFID Enabling Engine 確保世上任何物流點用戶均可透過網絡(網絡瀏覽器)登入系統進行相關操作。

附錄 A 資料庫規格

RFID Enabling Engine 之資料庫將會以 ER Diagram 形式顯示:

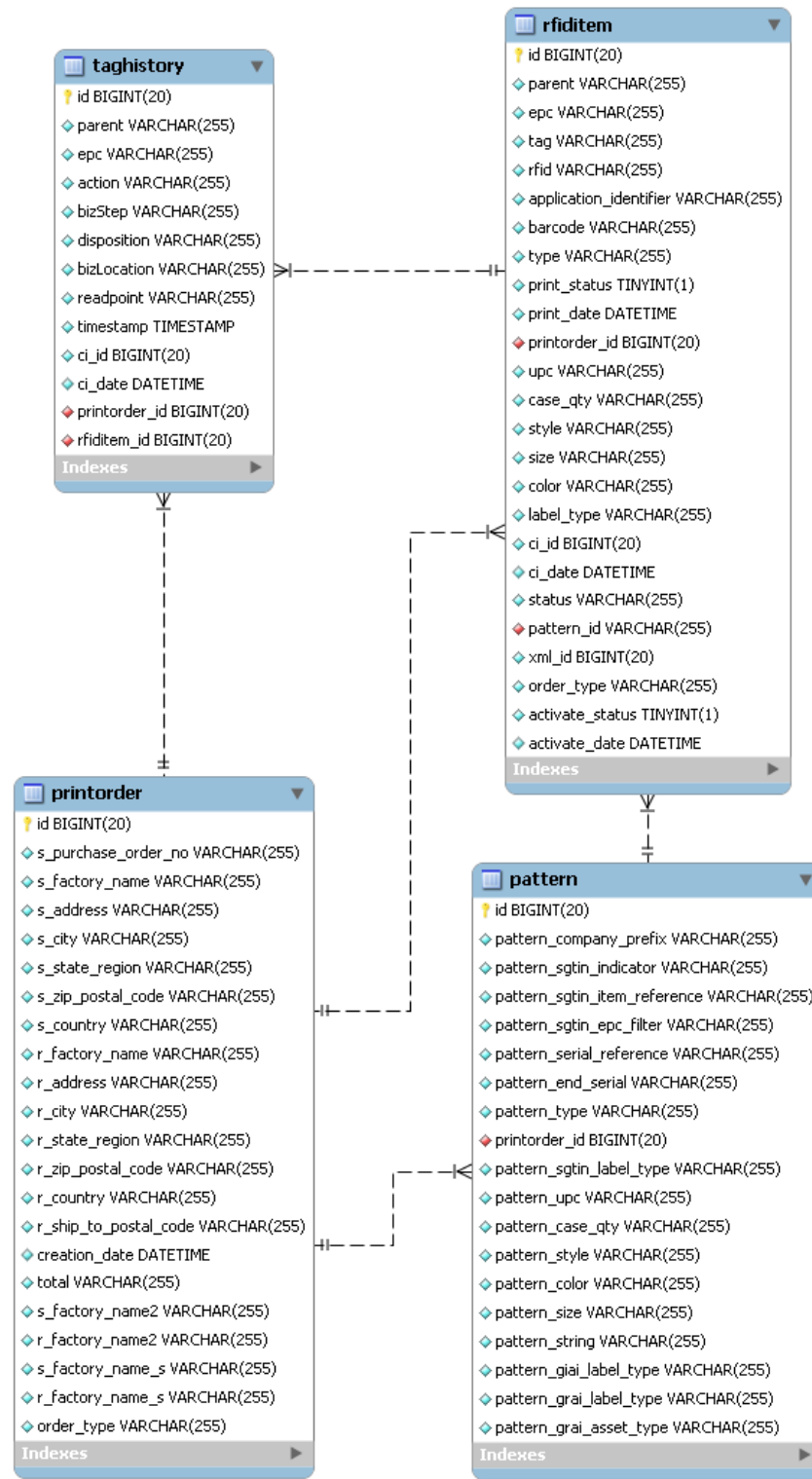
EPCIS 資料架構

以下為 EPCIS 事件資料之資料架構:



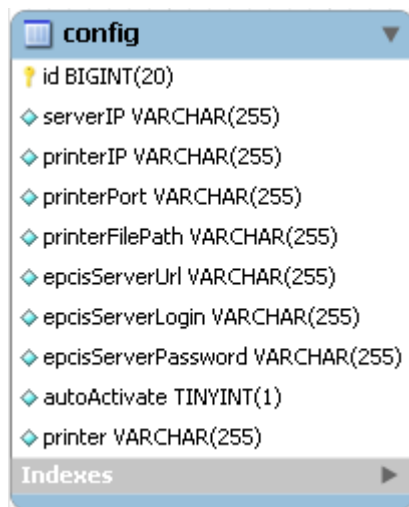
主資料架構

以下為主要功能模組之資料架構:



基本資料架構

以下為系統設定之資料架構:



Field Name	Data Type
id	BIGINT(20)
serverIP	VARCHAR(255)
printerIP	VARCHAR(255)
printerPort	VARCHAR(255)
printerFilePath	VARCHAR(255)
epcisServerUrl	VARCHAR(255)
epcisServerLogin	VARCHAR(255)
epcisServerPassword	VARCHAR(255)
autoActivate	TINYINT(1)
printer	VARCHAR(255)

Indexes

數據庫表 List:

數據庫表	Data Description
config	The config table is storing the system configuration.
epcis_datastore	The epcis_datastore 數據庫表 is storing the records which submit to EPCIS.
epcis_event	The epcis_event 數據庫表 is storing the records of EPCIS event.
pattern	The pattern 數據庫表 is storing the information for system internal use.
printorder	The printorder 數據庫表 is storing the record of each tag input.
rfiditem	The rfiditem 數據庫表 is storing all information of each tag.
taghistory	The taghistory 數據庫表 is storing the information sent from reader.

詳細表格說明:

1. config: The config table is storing the system configuration.

Field name	Data type	Length	Setting	Default	Description
Id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
serverIP	varchar	255	-	NULL	The server IP of middleware
printerIP	varchar	255	-	NULL	Printer setting
printerPort	varchar	255	-	NULL	Printer setting
printerFilePath	varchar	255	-	NULL	Printer setting
epcisServerUrl	varchar	255	-	NULL	EPCIS setting
epcisServerLogin	varchar	255	-	NULL	EPCIS setting
epcisServerPass word	varchar	255	-	NULL	EPCIS setting
autoActivate	tinyint	1	-	NULL	Activate setting
printer	varchar	255	-	NULL	Printer setting

2. epcis_datastore: The epcis_datastore table is storing the records which submit to EPCIS.

Field name	Data type	Length	Setting	Default	Description
ci_id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
epcis_event_id	bigint	20	unsigned, Not NULL	0	Foreign key – epcis_event
xmlcontent	longtext	-	-	NULL	The upload data to EPCIS in XML format
creation_date	datetime	-	-	NULL	The date and time of creation
submission_date	datetime	-	-	NULL	The date and time of submission to EPC network
updated	tinyint	1	unsigned	0	Submission status
count	bigint	20	unsigned	NULL	Number of EPC elements
printorderno	varchar	255	-	NULL	Batch No related
printorderid	bigint	20	unsigned	NULL	Foreign key – printorder
submit_url	varchar	255	-	NULL	Submission destination
submit_login	varchar	255	-	NULL	Login name to EPC Network
return_code	varchar	255	-	NULL	HTTP Code of EPC server return

3. epcis_event: The epcis_event table is storing the records of EPCIS event.

Field name	Data type	Length	Setting	Default	Description
eventID	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
specname	varchar	255	-	NULL	The internal system key, spec name of the event
bizstepname	varchar	255	-	NULL	The EPCIS key, business step name
action	varchar	255	-	NULL	The EPCIS key, action of the event
bizlocation	varchar	255	-	NULL	The EPCIS key, business location
readpoint	varchar	255	-	NULL	The EPCIS key, read point
dispositionId	varchar	255	-	NULL	The EPCIS key, disposition ID
template	longtext	-	-	NULL	Upload template file

4. pattern: The pattern table is storing the information for system internal use.

Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
pattern_compan y_prefix	varchar	255	-	NULL	Company prefix for generating RFID
pattern_sgtin_in dicator	varchar	255	-	NULL	SGTIN indicator for generating RFID
pattern_sgtin_it em_reference	varchar	255	-	NULL	SGTIN item reference for generating RFID
pattern_sgtin_ep c_filter	varchar	255	-	NULL	EPC filter for generating RFID
pattern_serial_r eference	varchar	255	-	NULL	Serial for generating RFID
pattern_end_ser ial	varchar	255	-	NULL	The end of serial for generating RFID
pattern_type	varchar	255	-	NULL	RFID generation type, SGTIN, GIAI or GRAI
printorder_id	bigint	20	unsigned, Not NULL	0	Foreign key – printorder
pattern_sgtin_la bel_type	varchar	255	-	NULL	The type of SGTIN, Carton
pattern_upc	varchar	255	-	NULL	UPC information
pattern_case_qt y	varchar	255	-	NULL	General information field
pattern_style	varchar	255	-	NULL	General information field
pattern_color	varchar	255	-	NULL	General information field
pattern_size	varchar	255	-	NULL	General information field
pattern_string	varchar	255	-	NULL	Generation result
pattern_giai_lab el_type	varchar	255	-	NULL	The type of GIAI, Pallet
pattern_grai_lab el_type	varchar	255	-	NULL	The type of GRAI, Container
pattern_grai_ass et_type	varchar	255	-	NULL	The asset type of GRAI

5. printorder: The printorder table is storing the record of each tag input.

Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
s_purchase_order_no	varchar	255	-	NULL	Batch number
s_factory_name	varchar	255	-	NULL	Sender address
s_address	varchar	255	-	NULL	Sender address
s_city	varchar	255	-	NULL	Sender address
s_state_region	varchar	255	-	NULL	Sender address
s_zip_postal_code	varchar	255	-	NULL	Sender address
s_country	varchar	255	-	NULL	Sender address
r_factory_name	varchar	255	-	NULL	Recipient address
r_address	varchar	255	-	NULL	Recipient address
r_city	varchar	255	-	NULL	Recipient address
r_state_region	varchar	255	-	NULL	Recipient address
r_zip_postal_code	varchar	255	-	NULL	Recipient address
r_country	varchar	255	-	NULL	Recipient address
r_ship_to_postal_code	varchar	255	-	NULL	Recipient address
creation_date	datetime	-	-	NULL	The creation time
total	varchar	255	-	NULL	The total number of tags
s_factory_name_2	varchar	255	-	NULL	Sender address
r_factory_name_2	varchar	255	-	NULL	Recipient address
s_factory_name_s	varchar	255	-	NULL	Sender address
r_factory_name_s	varchar	255	-	NULL	Recipient address
order_type	varchar	255	Not NULL	-	The internal system key

6. rfiditem: The rfiditem table is storing all information of each tag.

Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
parent	varchar	255	-	NULL	The internal system ID for association and Disassociation
epc	varchar	255	-	NULL	The EPC number of a tag
tag	varchar	255	-	NULL	The Tag number of a tag
rfid	varchar	255	-	NULL	The RFID hex number of a tag
application_iden tifier	varchar	255	-	NULL	Application identifier for GIAI and GRAI
barcode	varchar	255	-	NULL	The barcode of a tag
type	varchar	255	-	NULL	The type of EPC number
print_status	tinyint	1	-	0	//0=not printed; 1=printed
print_date	datetime	-	-	NULL	The date and time of printing
printorder_id	bigint	20	unsigned, Not NULL	0	Foreign key – printorder
upc	varchar	255	-	NULL	The UPC of a tag
case_qty	varchar	255	-	NULL	Tag information
style	varchar	255	-	NULL	Tag information
size	varchar	255	-	NULL	Tag information
color	varchar	255	-	NULL	Tag information
label_type	varchar	255	-	NULL	The level type //Item = item level; Carton = carton level; Pallet = pallet level; Container = container level
ci_id	bigint	20	unsigned	NULL	Foreign key – epcis_datastore
ci_date	datetime	-	-	NULL	Date of the foreign key
status	varchar	255	-	NULL	The internal system key
pattern_id	varchar	255	-	NULL	Foreign key – pattern
xml_id	bigint	20	unsigned	NULL	Reserved field
order_type	varchar	255	Not NULL	-	The internal system key
activate_status	tinyint	1	-	0	//0=not activated; 1=activated
activate_date	datetime	-	-	NULL	Date of activation

7. taghistory: The taghistory table is storing the information sent from reader.

Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
parent	varchar	255	Not NULL	-	The internal system ID for association and Disassociation
epc	varchar	255	Not NULL	-	Data from reader, EPC
action	varchar	255	Not NULL	-	Data from reader, action of the event
bizStep	varchar	255	Not NULL	-	Data from reader, business step name
disposition	varchar	255	Not NULL	-	Data from reader, disposition ID
bizLocation	varchar	255	Not NULL	-	Data from reader, business location
readpoint	varchar	255	Not NULL	-	Data from reader, read point
timestamp	timestamp	-	-	'0000-00-00 00:00:00'	Timestamp of record
ci_id	bigint	20	unsigned, Not NULL	0	Foreign key – epcis_datastore
ci_date	datetime	-	-	NULL	Date of the foreign key
printorder_id	bigint	20	unsigned, Not NULL	0	Foreign key – printorder
rfiditem_id	bigint	20	unsigned, Not NULL	0	Foreign key – rfiditem

附錄 B 用戶介面設計

根據之前不同觀點所述，以下為已定義好之用戶介面設計概念圖：

➤ 登入介面

RFID Enabling Engine 包含一登入介面以供用戶輸入 Username 及 Password:

The diagram shows a login interface for the RFID Enabling Engine. It features a light green background with the title 'RFID Enabling Engine Login Interface' at the top center. Below the title, there are three main components: a rectangular input field labeled 'Username', another rectangular input field labeled 'Password', and a rounded rectangular button labeled 'Login'. Each component is centered horizontally and has a subtle drop shadow effect.

Field Name	Description	Validation
Username	User login ID of the RFID Enabling Engine	tomcat-users.xml under Tomcat Installation Path
Password	User password corresponding to username	tomcat-users.xml under Tomcat Installation Path
Button Name	Description	
Login	Login RFID Enabling Engine	

➤ **資料輸入介面(SGTIN)**

資料輸入介面包含用戶輸入標籤資料介面以供往後打印標籤及核對之用。

以下為用戶輸入 SGTIN 資料之範例介面：

RFID Enabling Engine Information Input Interface

General Information

Reference No.

Sender Factory Name Sender Address

Recipient Factory Name Recipient Address Ship to Postal Code

SGTIN EPC Required Information

Indicator

Company Prefix Item Reference

Start Serial End Serial

Start Hex End Hex

EPC Filter UPC

Case Qty Color

Style Size

Middleware

Field Name	Description	Validation
Reference No*	Reference No. as defined by user (PO no., internal reference, etc.)	-
Sender Factory Name	Shipper Name	-
Sender Address	Address of Shipper	-
Recipient Factory Name	Consignee Name	-
Recipient Address	Address of Consignee	-

Ship to Postal Code	Ship to postal code	-
Indicator*	GTIN Indicator (select from 0 to 9)	0 to 9
Company Prefix*	Company prefix of corresponding item UPC	-
Item Reference*	Item reference of corresponding item UPC	-
Start Serial*	Starting serial no. for generating EPC serial	-
End Serial*	Ending serial no. for generating EPC serial	-
Start Hex	Starting EPC Hex for not using RFID Middleware conversion	-
End Hex	Ending EPC Hex for not using RFID Middleware conversion	-
EPC Filter*	EPC Filtering value defined by end user	0 to 3
UPC	UPC of the item	-
Case Qty	Case Qty of the item	-
Style	Style of the item	-
Color	Color of the item	-
Size	Size of the item	-
Middleware*	Select to connect RFID Middleware or not	-
*Mandatory field		
Button Name	Description	
Sample	Provide sample information of the input fields	
Add	Add a batch of labels for this particular reference no. (order)	
Next	Next to confirmation page and then save the record	
Reset	Clear all the filled field before Next	

➤ **資料輸入介面 (GIAI / GRAI)**

資料輸入介面包含用戶輸入標籤資料介面以供往後打印標籤及核對之用。

以下為用戶輸入 GIAI / GRAI 資料之範例介面：

Field Name	Description	Validation
Reference No*	Reference No. as defined by user (PO no., internal reference, etc.)	-
Label Type	Select Label Type (GIAI / GRAI)	-
Company Prefix	Company prefix of corresponding asset	-
Asset Reference	The starting asset reference	-
End Asset Reference	The ending asset reference	-
Start Hex	The Start EPC Hex	-

	(Compulsory manual input for not using Middleware)	
End Hex	The End EPC Hex (Compulsory manual input for not using Middleware)	-
Middleware	Select connecting to RFID Middleware or not	-
Button Name	Description	
Sample	Provide sample information of the input fields	
Add	Add a batch of labels for this particular reference no. (order)	
Next	Next to confirmation page and then save the record	
Reset	Clear all the filled field before Next	

➤ **標籤打印介面**

標籤打印介面包含用戶輸入搜尋資料(Reference No.)以尋找相關樣板/訂單。過濾後之搜尋結果會如範例介面顯示(標籤激活功能可設定為系統背後自動運作或如下圖般手動操作):

RFID Enabling Engine Print Tag Interface

Search Criteria

Search
Reset

Activate All

Reference No.	Print Delete
Reference No.	Print Delete
Reference No.	Print Delete
Reference No.	Print Delete

Field Name	Description	Validation
Reference No.	Reference No. as defined by user (PO no., etc.)	-
Button Name	Description	
Search	Search for Reference No.	
Reset	Reset for the input value of Reference No.	
Activate All	Manual activate all template	
Print	Selected order printing	
Delete	Selected order removal	

➤ **標籤關聯 / 分離介面**

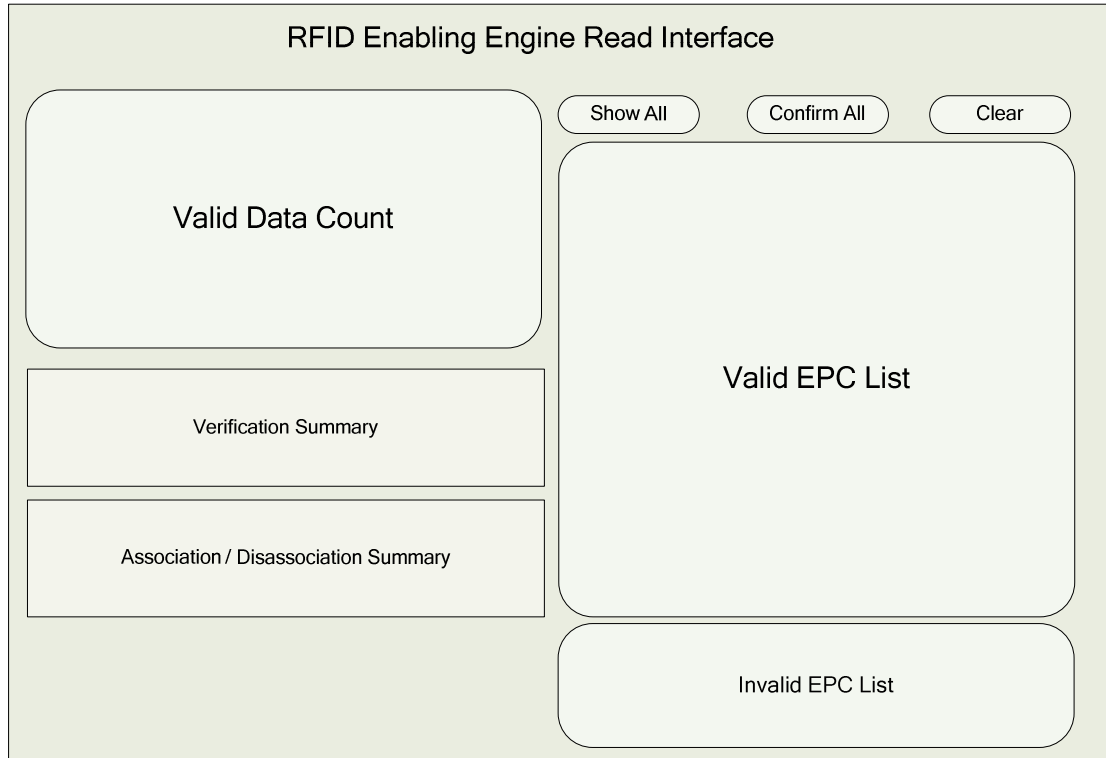
標籤關聯 / 分離介面包括 3 個主要部份: Data Count (左上), Associated / Disassociated EPC Summary (左下), Associated / Disassociated EPC List (右). Data Count 顯示由 RFID 手提讀取器接收到之有效的 EPC 關聯 / 分離資料。 Association / Disassociation 顯示有效之 EPC 關聯總數 / 分離總數。當用戶按 “Show All”，系統將顯示詳細之上層及下層標籤 EPC 資料。(系統會於背後核對標籤是否有效資料)

The screenshot shows the 'RFID Enabling Engine Associate Tag Interface'. At the top right, there are three buttons: 'Show All', 'Confirm All', and 'Clear'. On the left side, there is a 'Valid Data Count' section with two sub-sections: 'Association' and 'Disassociation'. On the right side, there is a large 'Valid EPC List' section and a smaller 'Invalid EPC List' section at the bottom right.

Field Name	Description	Validation
EPC	Parent's EPC (carton / pallet / container)	Valid in DB / Duplicate checking
EPC (Child)	Child's EPC (item / carton / pallet) of corresponding parent's EPC	Valid in DB / Duplicate checking
Type	Type of EPC (SGTIN / GIAI / GRAI)	-
Button Name	Description	
Show All	Show pop-up Parent EPC / Child EPC List	
Confirm All	Confirm all valid EPC association / disassociation data	
Clear	Clear EPC List	

➤ **標籤讀取介面**

標籤讀取介面為一總覽概要畫面，顯示由 RFID 讀取器已接收之 EPCIS 事件資料。Verification Summary 及 Association / Disassociation Summary 顯示核對 / 關聯 / 分離 EPCIS 事件資料總數。



Field Name	Description	Validation
EPC	EPC data with EPCIS event object	Valid in DB
Button Name	Description	
Show All	Show all valid EPC data	
Confirm All	Confirm all valid EPCIS event data	
Clear	Clear EPC List	

➤ **標籤核對介面**

標籤核對介面根據核對事件之定義顯示。有效記錄會顯示於 Valid EPC List:

The screenshot shows a web interface titled "RFID Enabling Engine Verify Tag Interface". It features three buttons at the top right: "Show All", "Confirm All", and "Clear". On the left, there is a box labeled "Valid Data Count". On the right, there are two larger boxes: "Valid EPC List" (top) and "Invalid EPC List" (bottom).

Field Name	Description	Validation
EPC	EPC (item / carton / pallet / container)	-
Type	Type of EPC (SGTIN / GIAI / GRAI)	-
Button Name	Description	
Show All	Show all valid EPC data	
Confirm All	Confirm all valid EPC data	
Clear	Clear EPC List	

附錄 C 製作 JAR 簡單指引

某些於程式中之 JAR (e.g. ALE_Schemas.jar , AleExt_xsd.jar , convert_schemas.jar, handheld_xsd.jar)是利用 Apache XMLBean 製作。以下步驟為一簡單指引 / 範例以展示如何製作 JAR 檔案(不同 JAR 檔均採用同樣步驟):

- 確認“ANT”之路徑已設定好於“System variable/系統變數”-“PATH”之中
- 同時確認“ANT_HOME”已設定好於“System variable/系統變數”
- 解壓“xmlbeans-2.5.0.zip”至“xmlbeans-2.5.0”
- 選擇“My Computer/我的電腦”，右按並選擇“Properties/內容”：
- 選擇“Advanced/進階”
- 按“Environment variable/環境變數”
- 尋找“System variable/系統變數”-“PATH”
- 按“Modify/編輯”並加上“bin”路徑例如：“...\xmlbeans-2.5.0\bin”於“Variable Value/變數值”
- 按“OK/確定”
- 於下方按“Add/新增”
- 於“Variable Name/變數名稱”輸入“XMLBEANS_HOME”
- 輸入 home 路徑如“...\xmlbeans-2.5.0”於“Variable Value/變數值”
- 按“OK/確定”
- 打開“...\xmlbeans-2.5.0\samples\XsdConfig”
- 刪除“xml”、“schemas”及“src”資料夾中之檔案
- 打開“build.xml”檔案
- 搜尋“catalog.xsd”
- 搜尋結果只有一個
- 修改<xmlbean schema="schemas/catalog.xsd"為<xmlbean schema="schemas"
- 複製目標 xsd 檔案至“schemas”資料夾 (註: 只需將 1 個 xsd 檔案移至資料夾)
- 打開 Window command prompt (於“Startup->Run /開始->執行”輸入“cmd”)
- 將路徑轉為“build.xml”檔案路徑
- 輸入“ant”command 並會於成功後顯示成功信息
- “build”資料夾會新增於路徑中
- 將路徑轉為“build/lib”檔案路徑
- 複製“schemas_xsdconfig.jar”檔案至目標路徑
- 重新命名目標名稱

附錄 D RFID Enabling Engine 及 EPCIS 附加資料

以下為參考資料:

Package: org.rfidee.web.action

Class: schedulingUpload2GS1

Package: org.rfidee.common.quartz

Class: processState

Package: org.rfidee.common.util

Class: HTTPPostSender

Package: org.rfidee.common.util

Class: SSLUtilities

通过 HTTP 上載的 Object Event 的參考 XML (樣品):

```
= <epcis:EPCISDocument creationDate="$creationDate" schemaVersion="1.0"
  xsi:schemaLocation="urn:epcglobal:epcis:xsd:1 EPCIS\EPCIS.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  xmlns:gs1hk="urn:epcglobal:gs1hk:xsd:ext">
= <EPCISBody>
= <EventList>
= <ObjectEvent>
  <eventTime>$eventTime</eventTime>
  <eventTimeZoneOffset>+08:00</eventTimeZoneOffset>
  <epcList>$epc</epcList>
  <action>$action</action>
  <bizStep>$bizStep</bizStep>
  <disposition>$disposition</disposition>
= <bizLocation>
  <id>$bizLocation</id>
  </bizLocation>
  <gs1hk:key01>$PO#</gs1hk:key01>
  <gs1hk:key02>$Upc</gs1hk:key02>
  <gs1hk:key03>$shipToPortalCode</gs1hk:key03>
```

```

<gs1hk:numeric01>$CaseQty</gs1hk:numeric01>
  </ObjectEvent>
</EventList>
</EPCISBody>
</epcis:EPCISDocument>

```

Reference XML (Sample) for uploading Aggregation Event through HTTP:

```

=<epcis:EPCISDocument creationDate="$creationDate" schemaVersion="1.0"
  xsi:schemaLocation="urn:epcglobal:epcis:xsd:1 EPCIS\EPCIS.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:epcis="urn:epcglobal:epcis:xsd:1"
  xmlns:gs1hk="urn:epcglobal:gs1hk:xsd:ext">
=<EPCISBody>
=<EventList>
=<AggregationEvent>
  <eventTime>$eventTime</eventTime>
  <eventTimeZoneOffset>+08:00</eventTimeZoneOffset>
  <parentID>$parent</parentID>
  <childEPCs>$epc</childEPCs>
  <action>$action</action>
  <bizStep>$bizStep</bizStep>
  <disposition>$disposition</disposition>
=<bizLocation>
  <id>$bizLocation</id>
  </bizLocation>
  </AggregationEvent>
</EventList>
</EPCISBody>
  </epcis:EPCISDocument>

```

附錄 E RFID Enabling Engine 及 DATAPLEX 附加資料

以下為參考資料:

Package: org.rfidee.web.action

Class (Method): PrintDispatchActionImpl.java

通过 HTTP 傳送到 DATAPLEX 的參考 XML (樣品):

```
= <Order xmlns="com.sedna.web.epcconvert">
= <EPCList>
= <SgtinItem>
  <indicator>1</indicator>
  <company_prefix>0037001</company_prefix>
  <item_reference>89012</item_reference>
  <serial_reference>123856</serial_reference>
  <serial_end_reference>123861</serial_end_reference>
  <epc_filter>3</epc_filter>
  </SgtinItem>
</EPCList>
</Order>
```

Reference XML (Sample) for receiving from DATAPLEX through HTTP:

```
= <Order xmlns="com.sedna.web.epcconvert">
= <EPCListResult>
= <Item>
  <barcode>10037001890122</barcode>
  <epc>urn:epc:id:sgtin:0037001.189012.123856</epc>
  <tag>urn:epc:tag:sgtin-96:3.0037001.189012.123856</tag>
  <rfid>3074024224B895000001E3D0</rfid>
  </Item>
= <Item>
  <barcode>10037001890122</barcode>
  <epc>urn:epc:id:sgtin:0037001.189012.123861</epc>
  <tag>urn:epc:tag:sgtin-96:3.0037001.189012.123861</tag>
  <rfid>3074024224B895000001E3D5</rfid>
  </Item>
</EPCListResult>
</Order>
```