
Trinity5

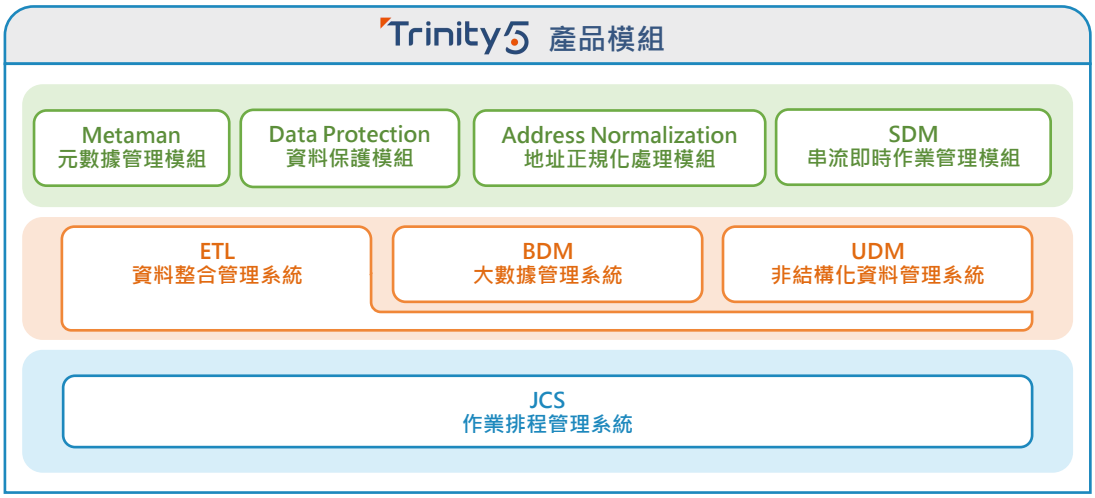
章節

關於 Trinity

作為國內 ETL 平台的先驅，NetPro Trinity 是業界首創結合工作排程與 ETL 管理，提供操作簡易、功能強大且高彈性架構的資料整合作業管理平台。本文將介紹 Trinity 運行技術基礎以及延伸應用，描述 Trinity 如何協助企業改善日常維運及設計工作，並克服作業管理過程之常見問題及挑戰，一舉提升資訊系統整合及運作的成效。

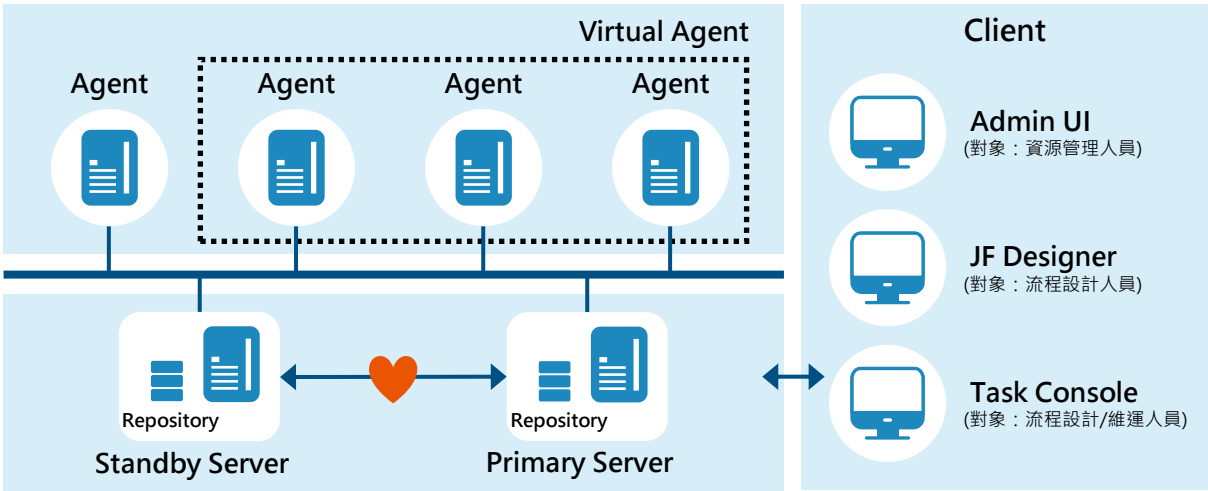
產品模組與系統架構

Trinity 是一套兼具資料整合與作業控管的管理平台。作為企業資料整合與商業智慧必備利器，透過工作排程管理、資料轉換管理、巨量資料管理、非結構化資料管理等資料管理與資訊整合模組，結合各式智慧應用，諸如資料保護、地址正規化、元數據管理與資訊治理等模組，打造企業資料資產價值，鞏固企業於資訊時代的競爭優勢。



分散式作業執行機制

TrinityETL 核心為分散式執行架構，包含一部 Server 以及 (至少一部) Agent。Server 負責所有作業的流程控制、排程啟動 (Scheduler)、條件式啟動、等待檔案、執行歷程記錄等監控及管理工作，並將系統資訊紀錄於其上方的 Repository 當中。
作業啟動後由 Agent 執行。Agent 可為一至數部，並且可由不同的作業平台及硬體規格組成。

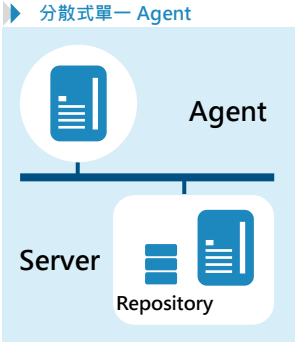


系統可擴充性

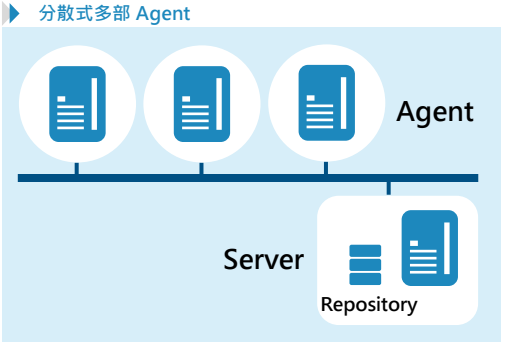
每一部 Agent 皆可同時執行多個作業，當企業作業數多或是資料處理量大時，只要橫向擴充 Agent 數量則可直接提升 Trinity 整體作業的運作效能。

單機版 Trinity

分散式單一 Agent



分散式多部 Agent



高穩定性架構

在 Server 的設計上，可部署 Server Redundancy 架構，當 Primary Server 因發生硬體故障或遭遇網路斷線等問題而造成作業無法繼續進行，Standby Server 就會迅速接手整體系統的運作。

備援及負載平衡

負責執行作業的 Agent 如有 2 部或 2 部以上，即可編組為「Virtual agent」(虛擬 Agent 群組)，群組內的 Agent 可以進行作業執行量負載平衡，且當群組中任何一部 Agent 故障時，群組內其他運作正常的 Agent 也會接手其工作，作業執行不中斷。

單一登入介面

Trinity 簡化登入介面，規劃了完整的單一入口，讓使用者在登入 Trinity 之後，不須重複輸入帳號密碼即可接續登入其他功能模組。市面上常見的 ETL 工具大多須由使用者透過程式呼叫不同工具模組元件後另行輸入帳號密碼才得以完成各項登入，相較於此，Trinity 簡化了系統操作流程，帶來更多便利。

Trinity Home單一登入介面

Trinity Command Center

若企業架設多套 Trinity，維運人員可以透過 Trinity Command Center 圖形化監控介面即時監控各 Server 及 Agent 的運行狀態及其上方所有執行中、等待中或發生錯誤的任務。維運人員直接點選此介面即可連結多套 Trinity，整合多套 Trinity 的管理，免去登入多部 Server 的手續，即可做到即時監控、即時回報與即時修正。

Trinity Home單一登入介面

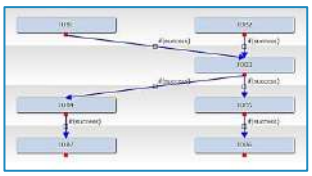
工作排程管理系統 (Job Control System，JCS)

TrinityJCS 可將複雜的資料整合工作以及精密的流程控管，透過易學易用的作業排程設計與管理介面，運用創新的分散式執行架構，經過簡單的設定，即可輕鬆進行設計及管理，使作業自動化執行，有效降低運維成本，提升 IT 營運績效。

作業流程管理

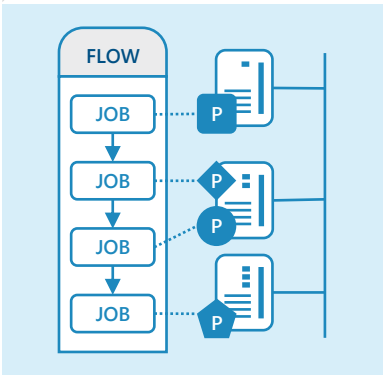
管理人員可透過圖形化介面進行作業流程設計及監控，並可藉由網頁瀏覽器，即時追蹤作業執行狀態。對於分散在不同硬體系統、不同平台上的作業，也可輕鬆設計各作業之間的執行順序與流向。所有作業的執行結果及 log 內容，亦集中於 Trinity 單一操作介面中完整呈現。

Job Flow 設計畫面



Job 監控畫面

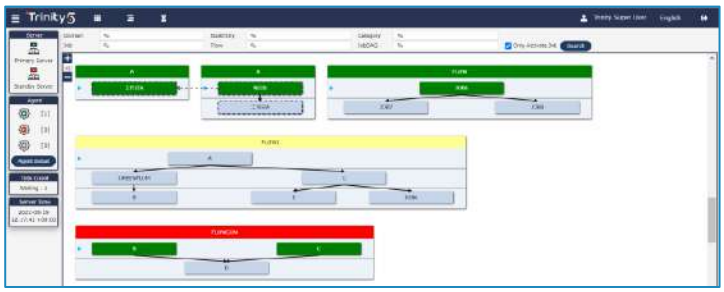
Job Flow 示意圖



作業協同機制

針對作業相依關係、作業的自動化執行順序、作業執行條件(Job Dependency)、自動觸發後續作業(Job Stream) 等流程情境，提供便利的結束狀態檢核條件設置及圖形化設計介面，以視覺化方式呈現設計結果。監控畫面亦呈現清楚的作業協同關係及作業等待原因，輔助日常維運及工作交接更加順暢。

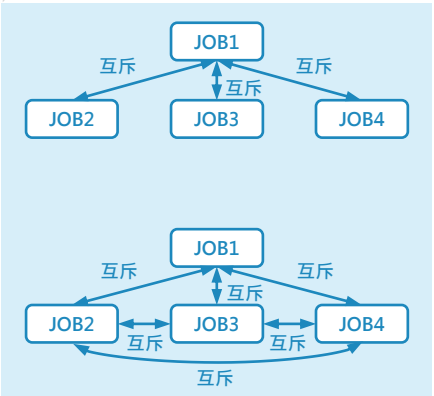
監控畫面呈現作業協同關係



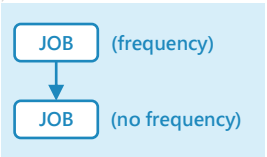
作業執行互斥設定

作業執行時，自動控管互斥作業不可同時執行。可設定一對多互斥或任兩個作業互斥。

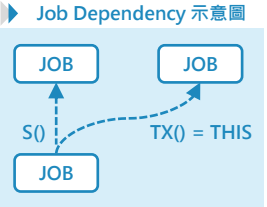
互斥情境



Job Stream 示意圖



Job Dependency 示意圖



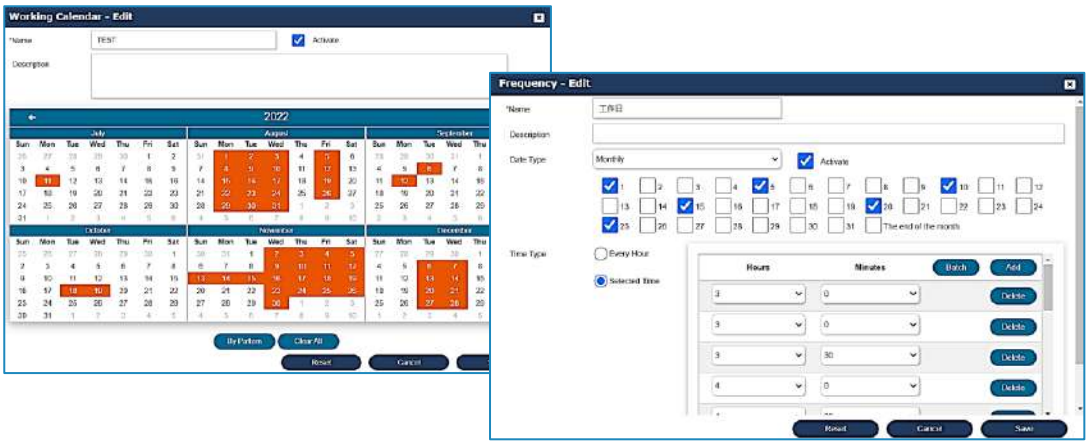
作業執行優先處理機制

除了既有排程時間的設置之外， 若有多組作業在 Trinity Agent 可同時執行的最大作業量範圍內同時間執行，可透過 Priority 優先順序來設定作業執行的先後次序。在作業執行時，於監控介面中也可對作業任務動態調整 Priority 優先順序。

可共用的排程條件規則

於圖形化介面建立排程條件規則，提供多種彈性的方式輔助使用者選取所需的日期及時間。已訂定的排程可提供所有作業共同使用。使用時只需於作業設計畫面上選用該排程，即可輕鬆套用。一旦系統時間符合排程條件，即可觸發作業自動化執行。

若發生例外情形則可建立例外排程，定義排程時間的除外時段，並可選擇套用在指定作業、作業流程或既有排程規則上，使得不須修改排程規則或作業設計內容，即可達到停機的效果。



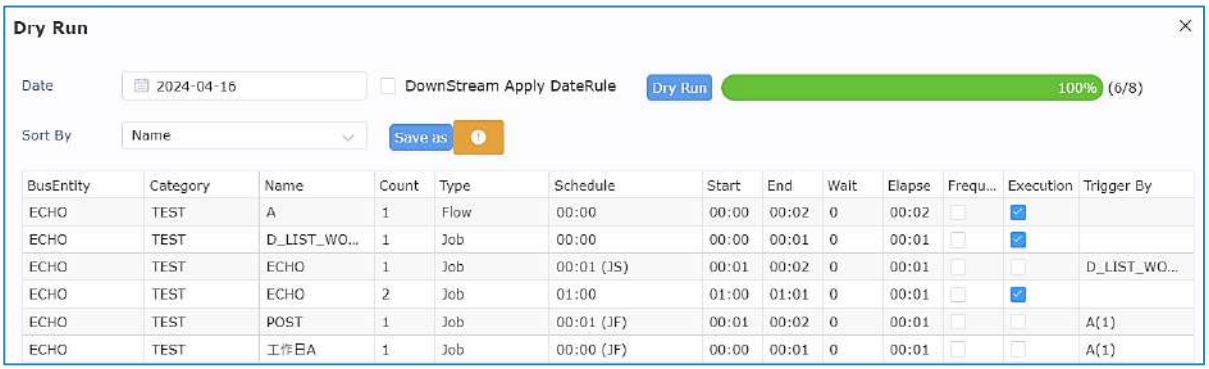
作業執行時程預覽

在作業的設計頁面具備執行時程預覽 Forecast 功能，可以方便開發人員預覽此作業套用排程條件之後的預計排程狀況，以利檢查作業排程條件是否正確無誤。

Job 設計畫面 - Forecast

作業演練 (Dry Run)

利用 Dry Run 功能可指定日期進行當日作業演練，將當日會預定的排程作業依序列出明細，並預估執行時間與完成時間。



BusEntity	Category	Name	Count	Type	Schedule	Start	End	Wait	Elapse	Frequ...	Execution	Trigger By
ECHO	TEST	A	1	Flow	00:00	00:00	00:02	0	00:02		☑	
ECHO	TEST	D_LIST_WO...	1	Job	00:00	00:00	00:01	0	00:01		☑	
ECHO	TEST	ECHO	1	Job	00:01 (JS)	00:01	00:02	0	00:01			D_LIST_WO...
ECHO	TEST	ECHO	2	Job	01:00	01:00	01:01	0	00:01		☑	
ECHO	TEST	POST	1	Job	00:01 (JF)	00:01	00:02	0	00:01			A(1)
ECHO	TEST	工作BA	1	Job	00:00 (JF)	00:00	00:01	0	00:01			A(1)

作業執行狀態即時警示通知

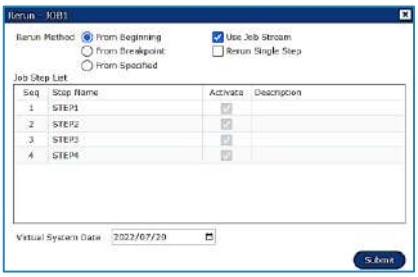
依據作業執行狀態，可設定系統自動發送訊息通知，以利維運人員即時掌握各個作業執行的情況。可選擇採用 SMS、E-Mail、HTTP、Command Line API 整合等方式通知使用者群組或是單一使用者，並可選擇是否將作業執行日誌以附件方式夾帶於信件當中。此機制可設定多種訊息發送時機，滿足維運人員各式監控需求。

失敗作業自動重新執行 (Retry)

透過將資料處理過程切割成不同的作業步驟，在作業執行失敗時便不需要將整個作業從頭來過。利用內建的「作業失敗重新執行機制」，可自動嘗試執行失敗的作業，直到執行成功或達自訂的重新執行上限次數為止。

另可指定重新執行時「從第一個作業步驟開始執行」、「從中斷點開始執行」、「從指定的作業步驟開始執行」等；這些重新執行的選項亦提供手動重新執行時選擇。

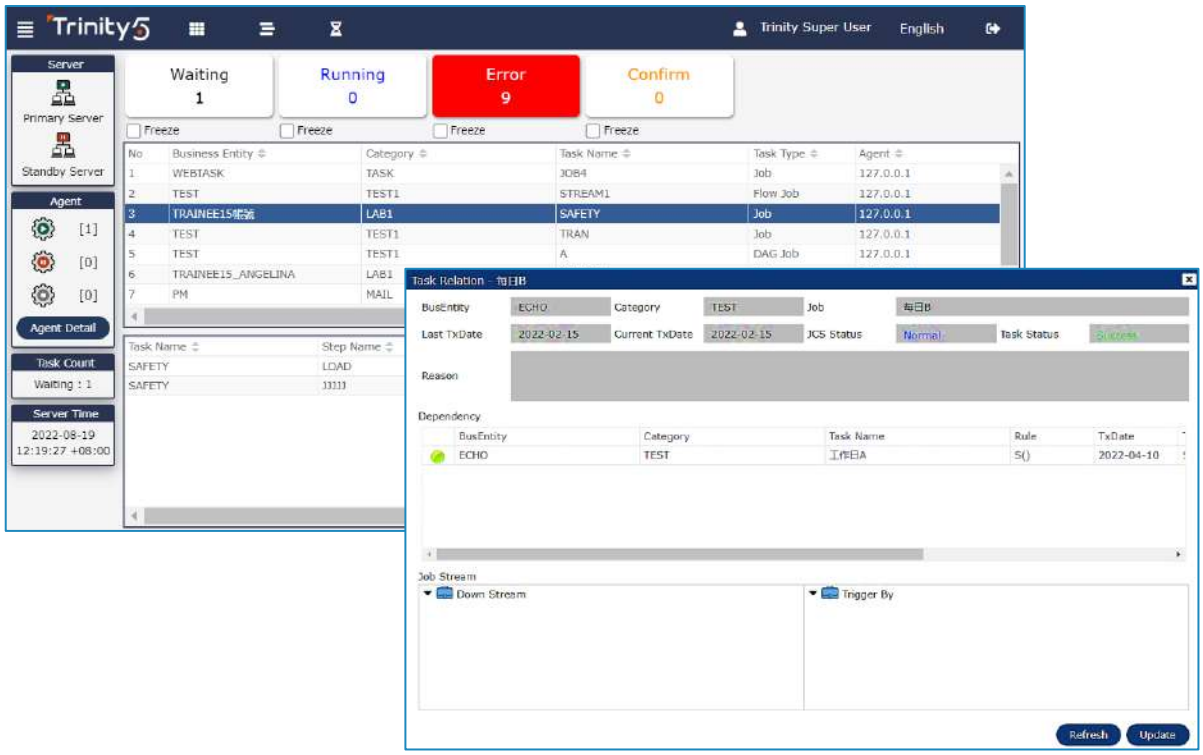
手動重新執行可選擇指定步驟



Job 設計畫面 - Retry Option

作業執行異常及排除

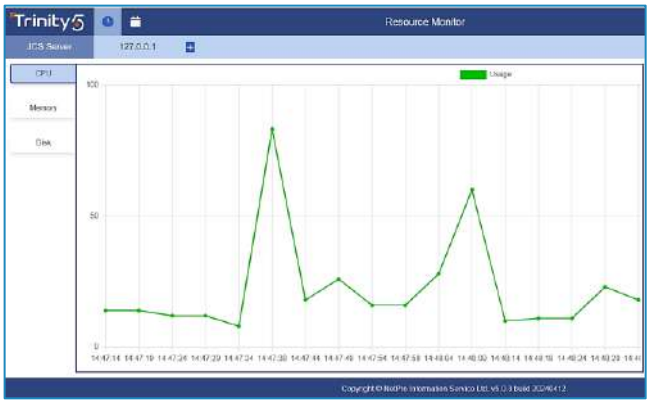
若作業執行異常將統一羅列，並呈現異常原因。若因作業依賴關係、檢核條件或作業優先權等因素造成該作業執行發生異常，可立即於維運介面採人為介入方式排除問題。



系統資源監控

可定義每台伺服器之同時執行作業上限數，以及定義硬體資源(CPU、Memory、Disk)的監控門檻；在資源使用不足時寄送警示訊息予維運人員。
此外，特殊設計的 WatchDog 機制，能嚴密監控所有 Server 及 Agent 的運作狀況，一旦發現問題，可於第一時間通知特定人員處理，縮短維運人員解決問題的時間。

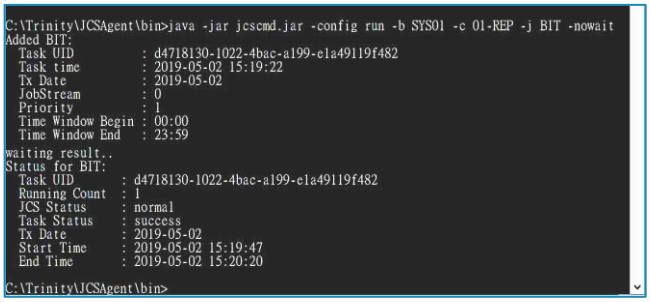
資源監控視窗



Command Line 方式執行作業

Trinity 支援透過 Command Line 的指令方式執行作業，包含提供 wait 或 nowait 參數，指示執行作業並等待執行結果或僅須執行而不需等待完成訊息；亦可指定作業執行的變數值與 Transaction Date，或指定使用 Job Stream 等多元化的執行方式。

資源監控視窗



Trinity API 與 Web Service

支援透過 Trinity API、Web Service、SOAP、REST 等介接方式執行資料讀寫或將資料發佈至指定目的地，滿足企業客製化 UI 或系統整合的使用需求。其中 API 的部份，企業只需使用任何一種可以呼叫 HTTP Request 的程式語言，如 Java、Javascript、.net、C、C++ 等，即可使用 HTTP Request 的 POST 或 GET 方式來呼叫正確的對應網址，輕鬆驅動後端來完成 Web Service 的呼叫。

完整的使用者權限控管

Trinity 擁有完整的使用者權限控管、角色權限控管及物件權限控管，不但於設定操作上相當便捷，亦可整合企業內部 Active Directory 或 LDAP 進行使用者身分認證，並可符合 CAS SSO 的作業模式，達到安全且便利的使用。

以參數化管理的連線資訊

Trinity 將連線定義作為共用資源集中管理，並以加密方式儲存使用者密碼，減少資安外洩的可能。而同組使用者帳號/密碼可以指定給多個使用者及作業共用，管理上更為便利。開發人員更可透過參數型式將連線相關資訊傳遞至作業程序當中，而不須將重要的資訊直接編寫在程式中。當連線資訊需要定期修改密碼時，僅需維護共用資源中的連線定義，則所有使用到該連線定義的作業將會一併生效，不須逐一維護作業。

強大的檔案接收機制

檔案接收機制支援以檔案名稱模式(File Name Pattern)作為定義，方便使用者對應多個格式相近的檔名。接收檔案過程會進行檔案數量檢核，使用者另可選擇啟用重複檔案檢核，及檔案完整性檢核(CheckSum比對)，確保接收檔案完整無毀損。收檔方式支援自地端取得檔案，也可經由 FTP/SFTP 檔案傳輸，自遠端來源取得檔案。

Trinity 將檔案接收機制作為共用資源集中管理。當檔案接收資訊定義完成後，透過作業選用此收檔規則，即可套用。後續如遭遇收檔路徑修改、檔名調整等情形，僅需維護共用資源中的檔案接收定義即可，不須修改作業。

內嵌外部腳本程式

當需要呼叫外部程式執行時，可直接於 Trinity 單一操作介面中編輯程式內容，不僅可進行版本控管，也不需因應多套 Agent 而維護多套程式，減輕作業開發人員的負擔；而外部程式執行結果亦可於監控介面保存並供查看，減少維運人員對於監控執行結果及管理執行紀錄上的困擾。

實用的任務執行條件設定

可依照作業特性，設定相同作業可同時執行，或作業執行時是否僅保留當前執行紀錄，或於執行前檢查同一作業的前次執行紀錄是否執行完成，若尚未完成 (如：等待中、執行中、暫停等)則不會執行新的任務。此外可選擇當系統運作異常時，是否忽略過去時段中缺漏的排程作業。

作業的版本控管

Trinity 版本控管機制可確保在一個時間僅有一個開發人員對單一作業進行維護，以避免不同開發人員互相覆蓋彼此的作業內容。透過建立正式版號可保留同一作業的多個版本，亦可運用「Rollback」恢復至指定版本，輕鬆達成版本控管需求。而運用內建的 Export/Import Job 功能即可將作業發佈至目的地作業環境當中。

檔案接取監控 (File Watcher)

除了批次作業模式及近即時 (near-real-time) 作業模式，Trinity 更具備「檔案接取」模式，可監控特定目錄的活動，一旦發現檔案傳遞完成則自動啟動相關作業。除了可監控單機目錄之外，也支援主動至 FTP Server 偵測並抓取檔案，並支援 FTP、SFTP、HTTPS 模式。檔案接取監控模式特別適用於檔案產生時間不固定、難以訂定排程的情境。

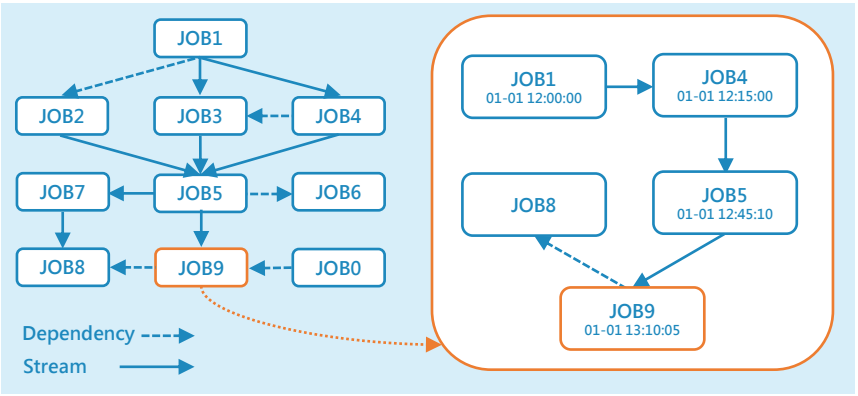
CAS SSO

Trinity 提供了符合「Central Authentication Service Single Sign On (CAS SSO)」的登入模式，可供使用者存取多個不同網路的應用程式時，僅需登入一次即可訪問所有相互信任的應用系統。對管理者而言，能更容易地管理系統，提升網路安全。

Job Critical Path Map 關鍵路徑

提供複雜的作業上下游路徑關係中，分析執行與最長所需之路徑進行預估關鍵作業的完成時間 。方便系統於忙碌或異常狀況時提供維運人員進行資源調整時的參考依據，簡化維運工作的負擔。可於作業串流中指定作業即時剖析其當前作業關係，找出最長完成的時間路徑以利釐清作業關係找尋其關鍵作業。針對已經存在的 Waiting Task 也可預估其完成時間 。

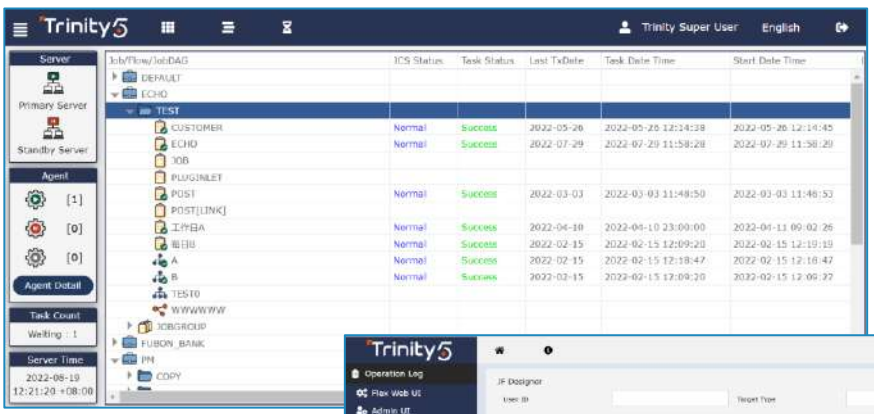
Job Critical Path Map 示意圖



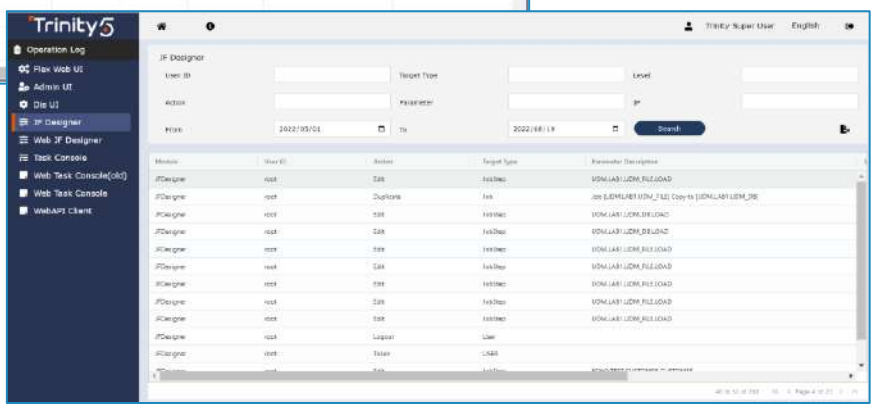
行動化、圖形化的作業執行/監控介面

系統監控人員可透過桌機或是平板電腦瀏覽簡潔明瞭的圖形化作業執行/監控介面，掌握日常作業的執行狀況、資料處理筆數及作業執行時間，包含作業執行紀錄與使用者操作紀錄也可輕鬆查詢。藉由不同顯示顏色的作業狀態呈現，監控人員對整體作業執行情形得以一目了然。而執行中的作業程序若需人為介入強制中止或手動重新執行，僅須透過在裝置上以單一畫面操作相關功能，即可達到目的。

作業執行紀錄



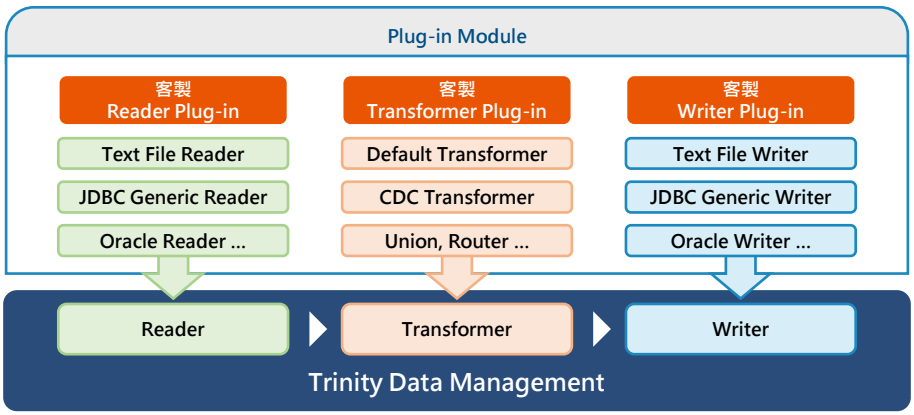
使用者操作紀錄



資料轉換管理系統 (Data Management，ETL)

TrinityETL 以圖形化介面提供豐富的參數設定及高效能的資料轉換功能，除了內建大量資料轉換、內碼轉換等函數，亦支援主流資料庫 Bulk Loader 與 CDC、JDBC/ODBC Multi-Threading 技術，可高速執行資料載入。結合 Trinity 工作排程管理，成為操作簡易、功能強大且高彈性架構的資料整合作業管理平台。

軟體系統架構



資料轉換系統包含了三個主要的部件，分別是資料讀取單元 (Data Reader)、資料轉換單元 (Data Transformer) 以及資料載入單元 (Data Writer)，功能說明如下：

- 資料讀取單元： 自資料庫或檔案等資料源擷取資料。
- 資料轉換單元： 將資料轉換為預訂的格式及資料型態。
- 資料載入單元： 將資料載入至目的地資料庫或產出檔案。

每個單元以內嵌元件 (plug-in) 的型式提供使用。其運作時，皆有獨立的執行緒 (thread)，並且使用多執行緒 (multithreading) 平行執行架構以達最佳的執行效能。因此不管是個別作業或是整個系統，其運作效能都能得到大幅提升，同時亦兼顧系統回應速度，使整個系統顯得既快速又靈活。

具擴充性的軟體架構設計

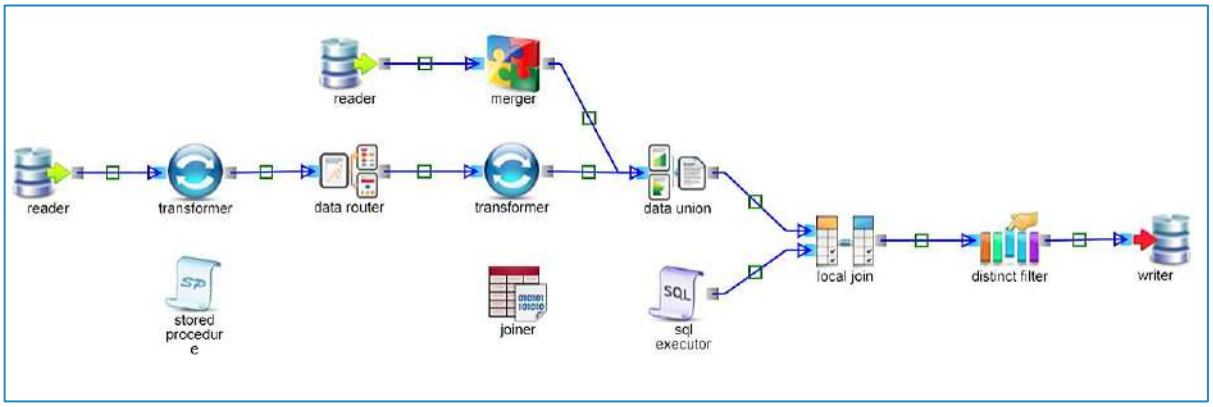
Trinity 由 Java 語言所構建，因此所有內嵌元件 (plug-in) 也繼承其延伸性，這也意味著未來只要是 Java 程式設計師，都可以依照預先設定好的規則，自行設計出具備特殊功能的內嵌元件；Trinity 亦支援外部程式呼叫執行，使企業系統應用變得更為彈性，更加符合企業各式各樣的需求。

豐富的資料轉換函數庫

Trinity 內建多種資料轉換函數，包含 DECODE、IIF、INSTR 以及 TRIM 等等，可滿足企業大部分的資料轉換需求，也提供由使用者自行定義轉換函數 (External Rule)。其他如資料的計算、合併、分流、連接、差異比對、彙總計算、資料轉置(正規化)，以及身份證及信用卡查核碼的檢核與產生、XML 資料解釋、內碼轉換等，都是支持企業高效完成工作的必備利器。

大幅優化的效能

Trinity 利用多執行緒 (multithreading) 技術讓每個作業平行處理，使作業環境維持高 CPU 使用率及高穩定性，快速完成大量工作。而面對 TB 等級的數據量，整合資料庫系統本身所擁有的批量載入工具 (bulk-loader)，使數據的載入能達最高效能；相對於傳統的 JDBC/ODBC 或是資料庫的驅動程式，Trinity 縮減了整體運作時程。



可自行擴充功能的 Groovy 腳本編輯器

Trinity 的 Groovy 元件可提供開發人員依照不同需求自行編輯腳本，解決複雜的特殊需求。此亦支援非結構化資料處理，以利企業解決變化多端的商業問題。同時在系統預設腳本檔案內已建立數個常用腳本，可縮短作業開發時間，是一個高效、敏捷且功能完整的模組。

靈活的 SQL 指令執行器

SQL 指令執行器除了可直接執行資料庫指令，還可加上邏輯判斷、迴圈、也可傳遞變數、設定錯誤機制處理等等，讓資料流程的安排更具彈性。

SQL指令執行器

SQL Statement

```
1   -- Trinity SQL Script
2   .SET echo 3
3   SELECT *
4   FROM task
5   WHERE startdate = '${TXDATE1}';
6
7   .IF (recordcnt !=0)
8       .goto STEP2 ;
9   .endif;
10  .print "No record. "
11
12  .quit 4;
13  .label STEP2;
14  .quit 0;
```

彈性靈活運用的python指令

除了運用 Trinity 的 Python 元件設計作業，亦可搭配排程直接執行既有的腳本。亦可透過 Trinity 的 Python Extract Transformer 執行 Python 腳本完成資料轉換的工作，該元件可視設計需求，宣告需要匯入的 libary，搭配 Python 腳本語法設計執行作業步驟內容。Python 腳本的支援可以因應使用者的設計習慣提供多種設計方式，以利解決變化多端的商業問題。

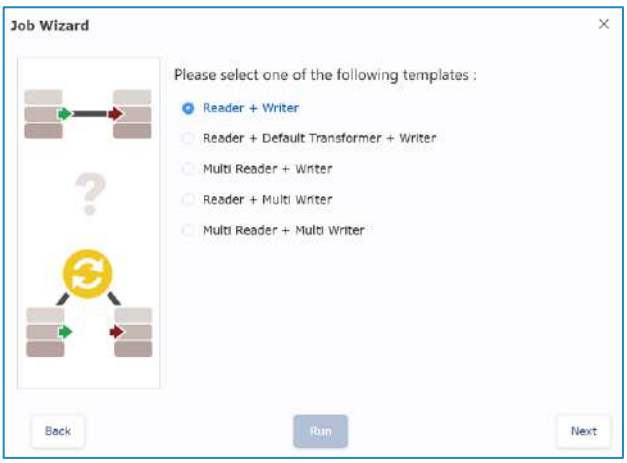
資料偵測處理與同步

透過組合、運用 Trinity 內嵌元件 (plug-in) 即可設計出各種資料轉換及資料讀寫需求的作業內容。舉凡資料同步、資料篩選、資料差異偵測、資料比對、條件判斷、異質資料處理及資料交換整合等，皆可輕鬆實現。

Job Wizard

依據元數據定義資訊，批次產生大量 ETL 轉檔作業。元數據範圍包含資料庫元數據及 Mataman 技術元數據。批次產製的設定過程相當簡單輕巧，按照引導步驟輸入設定值，即可輕鬆建立大量 ETL 作業。

Job Wizard 功能畫面



House Keeping

藉由 House Keeping 元件，可將系統執行過程所產生的暫存檔或 log 檔打包至指定目錄下或直接刪除，不須經過繁雜設定，即可輕鬆完成歷史資料檔案的維護工作。

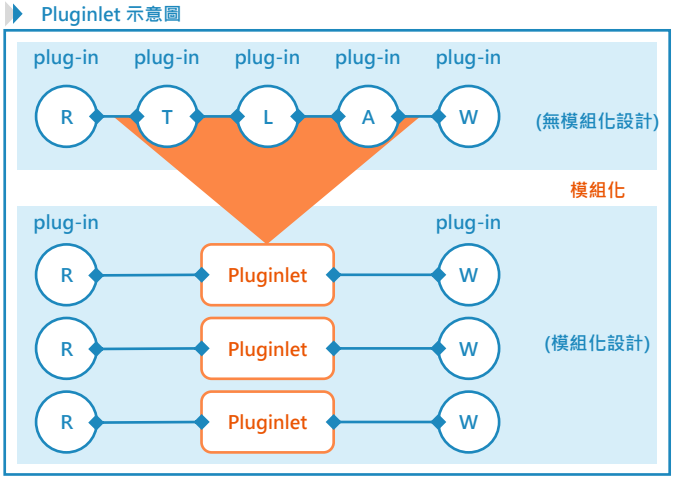
高性價比(Cost/Performance, C/P Ratio) 的 CDC

差異資料載入 (Changed Data Capture，CDC) 元件可針對同一來源的數據在不同時間點的數據量進行比較，並找出從上次執行的基準點後的異動情形，包含新增、更新、刪除等。這能讓資料量大但更新比率小的作業，大幅節省作業執行資料更新的時間。

模組化的作業開發方法

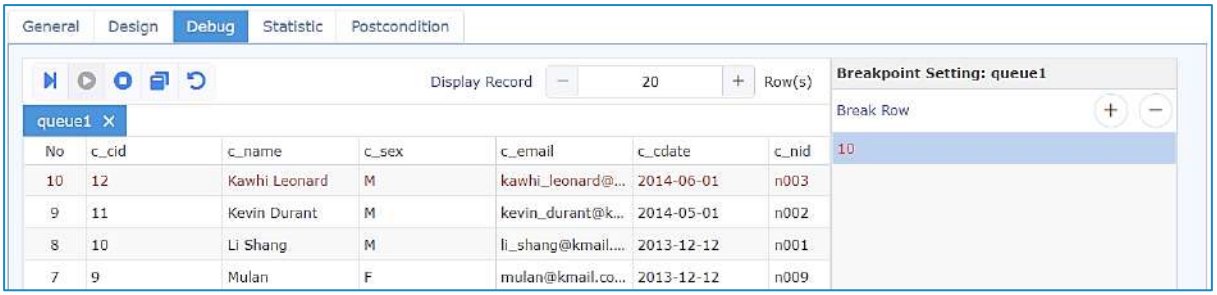
若多資料轉換處理作業的設計有相同或相似的部份，則可運用 Pluginlet (模組化內嵌元件) 的設計型態，將相同或相似的部份事先設計為一個 Pluginlet 單元，在於後續建立新作業時引用，即可簡化新作業的設計工序，快速完成多個作業的建立及維護。

Pluginlet 示意圖



開發除錯模式

開發除錯模式 (DM Debug) 能在設計資料轉換作業的當下協助開發人員快速偵測出不符合規則的資料列，縮短測試時間。

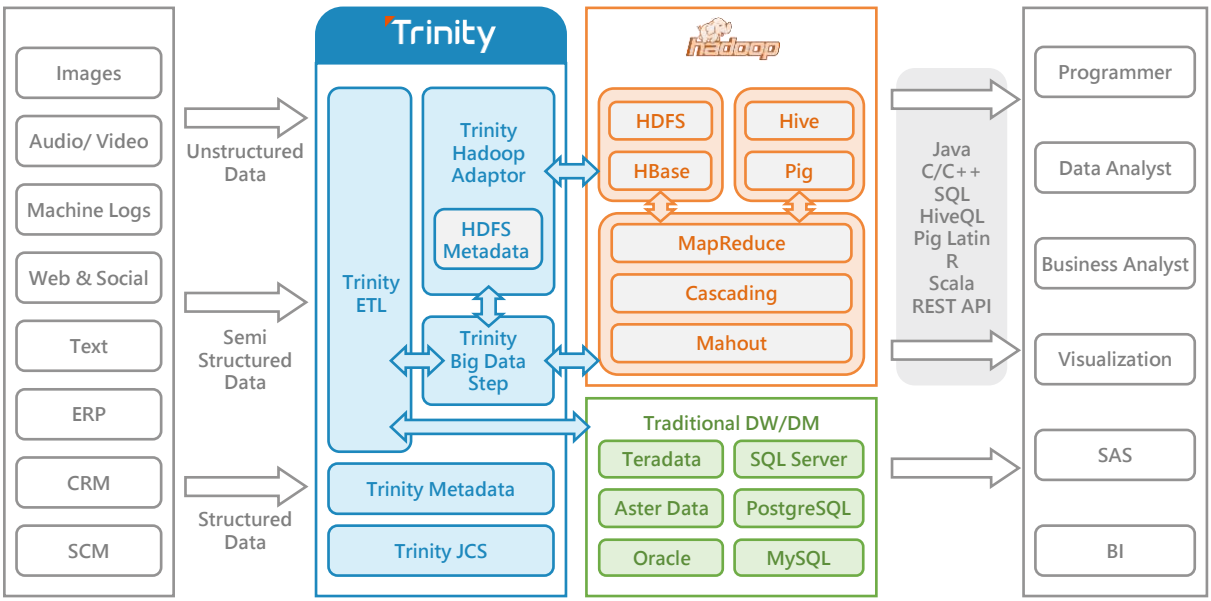


巨量資料管理模組 (BigData Management，BDM)

TrinityBDM 可自數據池無縫進行資料交換。資訊分析人員可以透過熟悉的工具介面進行操作，來取得這些數據源的集成，使企業大幅縮短 Big Data 應用的評估與導入時程，有效降低後續系統維運所需投入之成本及人力。

連接傳統ETL作業與Hadoop架構之間的數據交換橋梁

Trinity 透過其獨特的組件擴充架構，開發出 Big Data Adaptor，將 Hadoop 巨量資料的處理能力與傳統 ETL 作業有效且無縫地融合在一起。使用者對 Hadoop 不須具備高深的管理技術或程式開發技巧，即可在熟悉的 ETL 系統上輕鬆介接 Hadoop 相關的應用處理功能。不但既有系統的投資獲得保障，同時又能擁有 Hadoop 所帶來可靠、高效、可伸縮的巨量資料處理能力。



Trinity 當中提供專用的作業步驟，以元件化方式提供使用者定義 Cascading Multitool、MapReduce Operation、HDFS delete Operation、HDFS Mkdir Operation 等操作，結合 Trinity 共用資源統一管理的特性及 TrinityJCS 作業排程管理功能，強化使用者在 Big Data 作業的設計及維運流程。

簡化對 HDFS 上資料讀取與載入

Trinity 提供 HDFS 的讀取組件及載入組件，可直接對 HDFS 上的檔案進行讀寫，亦提供 HDFS 目錄結構的瀏覽功能，方便開發人員快速完成作業設計。

簡單的 MapReduce 程式執行介面

Trinity 提供了簡單易用的圖形化介面，使開發人員可以透過簡易的設定，執行既有的 MapReduce 程式，省略繁雜的發佈程序，降低開發人員應用 MapReduce 的門檻。

整合 Hbase 中的結構化資料

對於已經存在於 HBase 上的結構化資料，可直接透過 Trinity 所提供的 HBase 讀取元件進行取用；HBase 載入元件也支援將一般資料庫或文件中的結構化資料載入至 HBase 當中以進行開發所需的分析。

整合 Hive 與 Pig

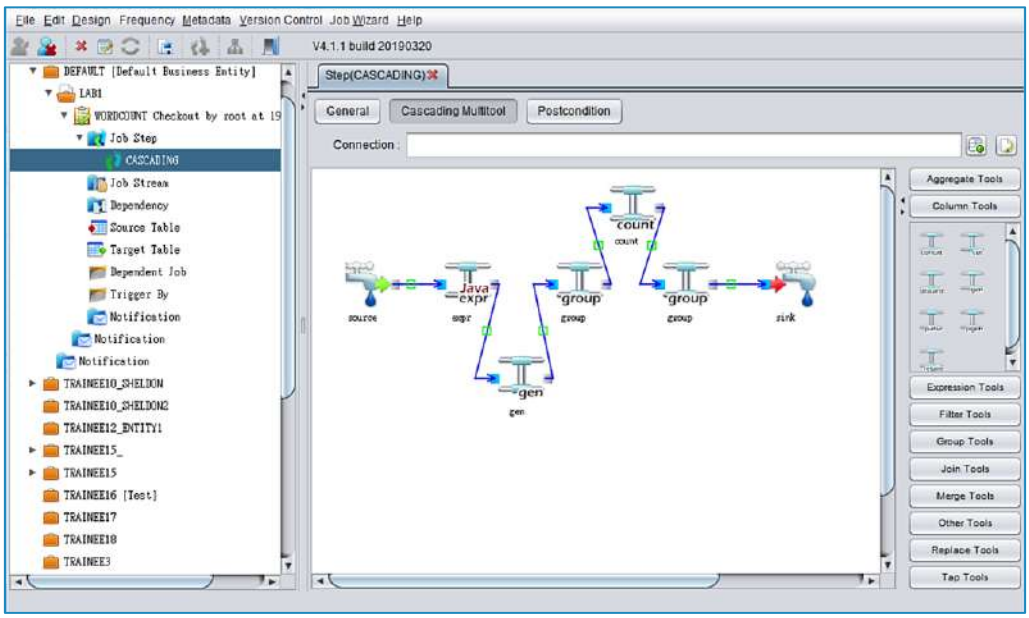
Trinity 提供 Hive 與 Pig 語言的執行組件，可供開發人員編寫複雜的程序以應用於 Hadoop 上的數據分析工作，並可將執行結果傳遞至 Trinity 當中的其他組件，無縫結合 Trinity 資料轉換管理及資料分析功能，達成 Hadoop 程序與 ETL 作業間數據交換的目標。

Mahout

利用 Trinity 中的 Mahout 組件，能讓開發人員更方便使用 Mahout 中所提供的各項演算法。藉由內建範本，開發人員可以直接呼叫使用 ，大幅減輕撰寫程式的負擔。

Cascading

開發人員利用 Trinity 中的 Cascading 組件， 可以更加輕鬆、快速地利用內建 Function 處理資料分析，簡化複雜的程式。

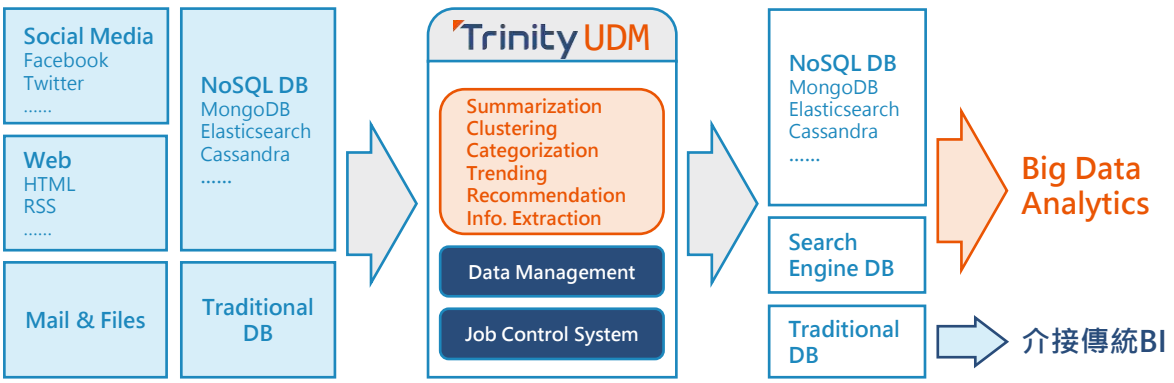


非結構化資料管理模組 (Unstructured Data Management，UDM)

TrinityUDM 內建非結構化資料分析引擎，以及存取、轉換、處理等相關組件，具備文本資料探勘功能以及 NLP(Natural Language Processing) 等演算法支援，可將非結構化資料轉換為結構化資料，使其與企業既有資料倉儲及分析工具進行整合。

非結構化資料管理

非結構化資料的價值已經大幅受到企業關注。運用非結構化資料管理系統，在企業確立了一個清晰的願景之後，將可對網路上分散的大量潛在資訊進行更為集中的收集及分析。

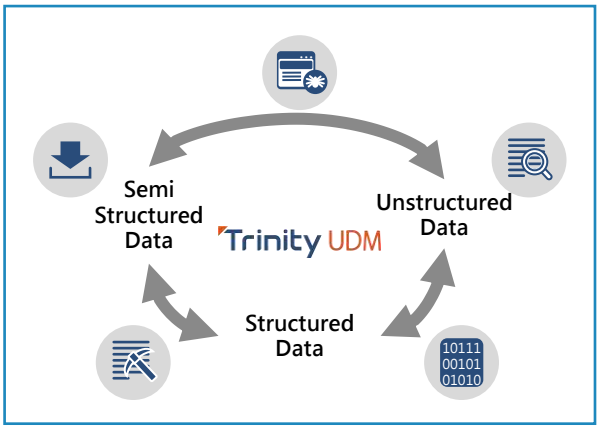


網路爬文與 Open Data 多樣化資料讀取

TrinityUDM 提供網路爬文 (HTML/RSS 等各種網頁、與 Facebook/Twitter 等社群媒體)、各式檔案(MS Office/Open Office/PDF/RTF/Epub)讀取 、主流 NoSQL (MongoDB/OrientDB/Cassandra) 與搜索引擎 (Solr/Elasticsearch) 資料庫讀取。

豐富的文本分析處理

針對所讀取的文本內容，TrinityUDM 提供語系識別、文件斷句切詞、文件相似性判斷、文件分群/分類、文件摘要、文件主題/趨勢等分析處理功能，甚至可將文本內容抽取，轉換為結構化資料，並且充分發揮在地的中文優勢，提供外商難以企及的分析應用。



供應 Big Data 與傳統 BI 分析應用

企業可快速處理內部文件與外部網路等非結構化資料；前者可作為合約比對處理、文件歸類管理等應用，後者可作為網路意見調查、網路知識訂閱、網路輿情收集、Open Data 資料擷取等應用。將擷取結果整合至內部文件或資料庫當中，再透過 Trinity 組件進行分析或儲存。

功能說明

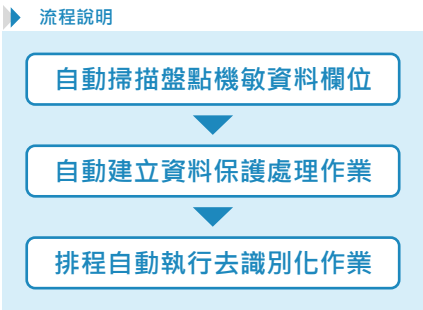
功能	說明
詞庫式分類	將中文文章依照指定詞庫分類
文章相似度計算-自動摘要	文章切句後，計算與原文相似度最高前幾句
文章相似度計算 & k-NN分類	給定一個資料來源及待測欄位，評定此欄位與資料來源各個實例的相近程度，並依據 k-Nearest-Neighbors (kNN) 演算法推薦最接近類別。此方法容易推測結論，能運用在任何資料型態、甚至是非關聯式資料上。但其空間複雜度高，計算量大，且特徵的選擇會與推測的精確性高度相關。
正負評論計算	使用文法規則及詞庫，判斷文章正負值。
意見維度	使用 latent Dirichlet allocation (LDA) 將所輸入的句子計算後輸出主題維度，利用主題模型可以挖掘文本資料背後的語意，適合用於文本分類、話題檢測、關聯判斷等。
斷字詞頻	使用Open Source: HanLP: Han Language Processing + 詞庫式合併新詞
斷字切詞	使用Open Source: HanLP: Han Language Processing + 詞庫式合併新詞
斷詞合併	依照使用者選取範圍資料，將斷詞合併加總詞頻
新詞	選取資料並且合併成一大篇文章，再用找新詞元件推薦新詞
斷詞詞性	使用Open Source: HanLP: Han Language Processing 判斷詞性
聚合式階層分群法	此元件內部使用聚合式階層分群法 (agglomerative hierarchical clustering)實作。其原理是先將每一筆資料視為是一個分群，之後逐漸將相似的分群合併在一起，一直到收斂條件達到時便停止。

資料保護模組(Data Protection)

資料保護已經是所有企業責無旁貸的重要工作，2018年歐盟更是祭出號稱國際間最嚴格的資料保護規範GDPR，金融業、航空業，以及電子商務業等皆會受到極大影響。Trinity 協助企業掃描/盤點資料庫當中的機敏資料，對其進行去識別化作業，保護企業資料在傳遞過程免受機敏資料外洩或觸法侵權的威脅。

全自動化流程設計

由 Trinity 掃描資料庫一般欄位及 XML/JSON 格式資料，盤點得到企業所有機敏資料欄位。再依據掃描結果，自動建立所有資料保護相關作業。過程當中不須仰賴人工建立，且能處理資料庫特殊情況，例如 constraint、foreign key、Blob/Clob, identity insert、..... 等。



便利的企業資料掃描

由企業自行定義機敏資料原則，再由 Trinity 自動掃描企業資料庫及資料表，盤點所有機敏資料欄位，並以文字檔紀錄需要進行資料保護的目標，供後續啟動資料保護相關作業時參照使用。

安全的資料處理過程

自資料庫讀取資料的過程中即進行資料保護，有效防止洩漏加解密方式與金鑰，確保資料保護措施不會被破解。

彈性的資料保護作業設計

企業可設定使用遮罩、加密、加簽或格式轉換等資料保護方式，及自訂去識別化規則，例如字串的遮罩或保留、文字格式/日期時間/Json/Xml 等格式轉換、自訂演算法與金鑰加密(MD5、DES)、關鍵值資料(身份證號、信用卡號) 加簽等。

關鍵值加簽後仍為合法資料

內建的資料檢查碼機制，能在進行資料去識別化時，自動產生合法的檢查碼，諸如身分證字號或信用卡卡號的檢查碼。可使資料在進行保護處理之後，不會因為前端 AP 系統的檢查機制而將其視為不合法的資料。

地址正規化模組 (Address Normalization，AN)

TrinityAN 可將戶政司全國門牌地址及中華郵政地址郵遞區號進行整合，打造出企業的「標準地址資料庫」，有效幫助企業對既有地址資料或新進資料自動進行校正，大幅節省人工比對地址的時間成本，利於企業維護資料正確性，進而提升資料分析準確度，確保行銷觸及範圍達到預期。

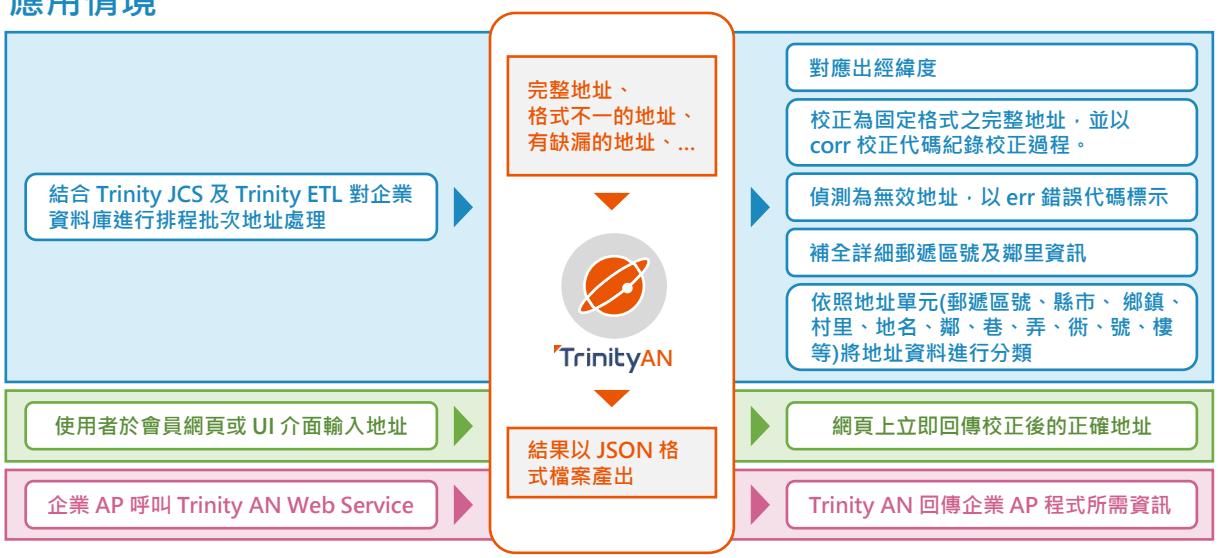
根絕地址錯誤

在眾多類型資料的處理當中，臺灣中文地址資料屬於變更極為頻繁的資料類型。地方機關的地名與路名不斷異動，戶政司資料頻繁更新，皆會造成企業地址資料發生資料缺漏、資料不完整、編碼不符、字型轉換不一致、夾帶特殊符號、資料格式不統一、……等多項問題。TrinityAN 可整合戶政司全國門牌地址及中華郵政郵遞區號，建立「標準地址資料庫」，提供企業比對校正既有地址資料，使企業資料的準確度與國家資料同步。不但能提升客戶服務品質，亦能大幅降低錯誤寄送信件或遭遇退件的問題及成本，促進有效的行銷及良好的發展。另可延伸融合地理資訊圖資雲服務平台(TGOS)的經緯度資訊，將地址定義增添經緯度對應，拓廣地址校正結果的應用範圍。

服務目的

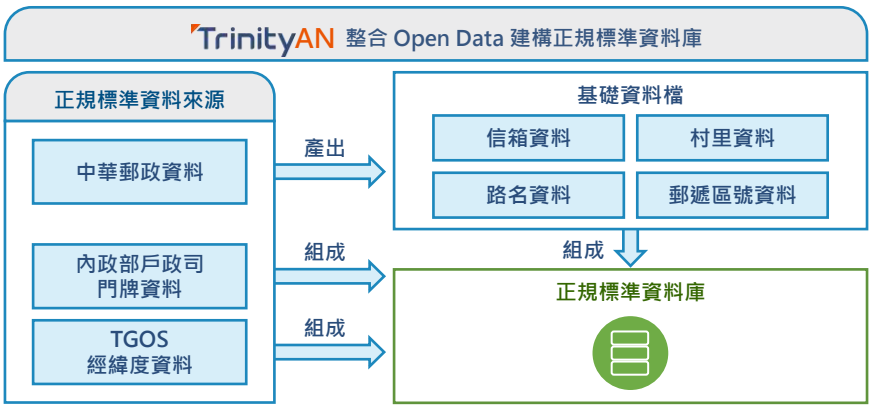
- 將傳入之地址以郵政資料做地址切割，並使用內政部戶政司門牌資料進行資料驗證，回傳校正後之地址字串。
- 回傳五碼郵遞區號、縣市、鄉鎮、村里、地名、鄰、巷、弄、衖、號、樓等每一個地址單元，可供企業進行運用。
- 紀錄校正過程與比對錯誤原因，並輸出標準資料庫比對結果。錯誤原因以錯誤碼表示，可供企業用於地址資料的分類或應用。

應用情境



建造企業專屬的正規標準地址資料庫

各單位所提供的 Open Data 更新速度不一，資料的範疇與格式亦不相同。運用 Trinity 設計作業流程，定期自動統合各單位 Open Data 資料，維護一個企業專屬的正規標準地址資料庫，使地址校正作業能夠快速且精準地完成。



完整的比對及校正

- 可進行模糊缺漏補回，包含鄉鎮市區、村里鄰補回。
- 可將地址字串處理成為各個地址單元(地址格式分欄)。
- 與全國門牌資料比對地址的正確性。
- 透過中華郵政資料取得郵遞區號(3+2)進行整合。
- 透過TGOS取得經緯度資料與企業地址資料進行整合。
- 比對及校正結果以 JSON 格式產出
  - 無效的地址會於 err 欄位以錯誤碼表示，代表地址不存在或無法辨識，屬於嚴重錯誤碼。
  - corr 欄位標記校正代碼，紀錄此筆地址正規過程的行為；此欄位可為多個，結構為一個極短且易分辨的數字陣列。

維運簡單易行

- 單一元件即可完成 Open Data 下載與更新標準地址資料庫。
- 可透過 Trinity 平台功能執行臨時的比對校正工作。
- 提供 API，可與企業各個 AP 進行資料交換處理或整合。
- 可規劃為 Server Cluster 備援機制，以虛擬 IP 連結，使 Web Service 可自動切換。當主要主機無法運作時，及時切換至備援主機，且影響 AP 端十分輕微。

元數據管理模組 (Metaman)

Trinity Metaman 支援企業達成資訊治理(Information Governance)，以高質量的資訊溝通體系使企業更易於掌握趨勢與影響力。

元數據管理

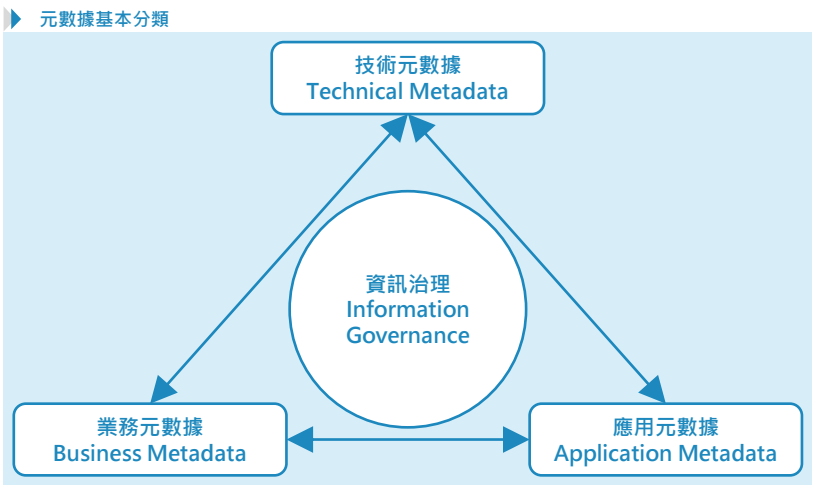
元數據是描述資料/定義/系統的說明資訊，大量的企業資料和職能知識可透過元數據加以表達。被良好管理的元數據可被企業人員更加有效地運用，讓這些寶貴的資源發揮於職能發展及工作任務當中。元數據管理亦是任何一個分析應用系統建置的重要基石，當資訊採用一致的標準、格式及定義來表達，不僅可使營運分析工作獲得較為準確的結果，也有助於降低資料交換及資料整合的成本，進而提高客戶服務品質。

權限控管方面則獨立於 Trinity 角色權限設定之外， Trinity Metaman 使用者角色權限區分為管理者、核准者、瀏覽者、編輯者及發佈者等角色。可將 Trinity 使用者帳號加入至角色的成員中以繼承相關權限。除了系統角色之外，亦可增加自訂角色再給予所需的權限許可。

元數據基本分類

技術元數據(Technical Metadata)
用來管理實體資料庫中的欄位型態定義資訊，包含實體屬性及DDL 定義。支援主流資料庫元數據之擷取與差異比對，並可進行數據剖析(Data Profile)及數據質量(Data Quality)工作。可建立欄位關聯，產出血緣關係表。

業務元數據
(Business Metadata)
用來定義各行業特定的業務名詞、業務規則、業務情境以及代碼表等資訊。業務元數據可使企業全體獲得一致的業務理解，促進有效的交流與溝通。



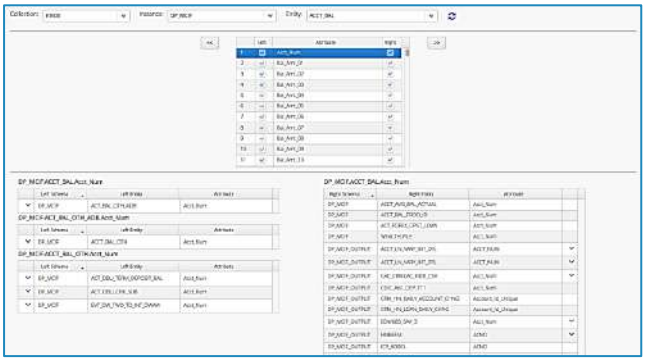
應用元數據
(Application Metadata)
用來定義各項系統/程式/報表的定義資訊。配合業務元數據及技術元數據，可完善企業知識管理(Knowledge Management)。此外，亦整合 ETL 元數據管理於此模組當中，支援 ETL 作業設計。

元數據血緣關係

元數據血緣關係大致可分成三類：

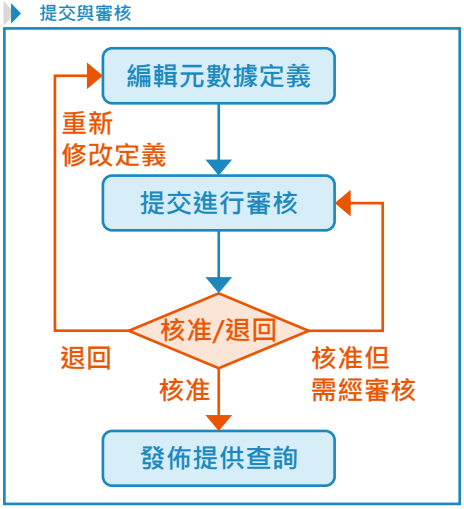
- 資料庫表之間的來源與目的關係
- 資料庫表與 ETL 處理程式間的關係
- 資料庫表與應用程式或報表程式間的關係

血緣關係表格



資料表血緣關係不同於 ER-Model 關係，後者僅指出表之間的 Primary Key/Foreign Key 參照關係；資料表血緣關係包含表中每個欄位的資料來源及資料去向，這些資料關係通常存在於作業設計內容、SQL Script 程式邏輯或執行日誌(例如：Teradata BTEQ執行日誌)之中。Trinity Metaman 可解析這些內容並呈現出血緣關係表格或血緣關係圖。後續可進行衝擊分析(Impact Analysis)。

提交與審核



技術元數據與資料庫之連結

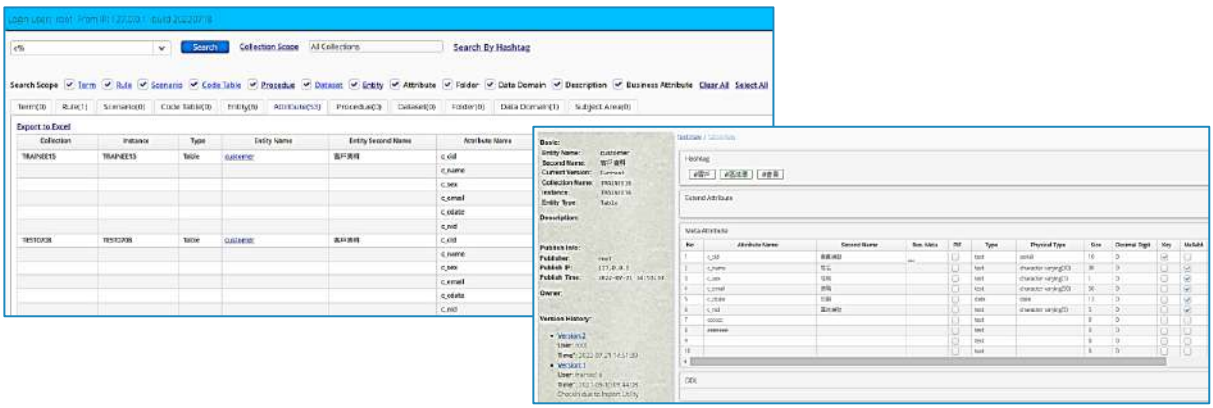
技術元數據代表一組資料定義的集合，除了以檔案匯入的方式編輯資料定義，亦可透過直接連結資料庫的方式將資料定義匯入至 Trinity Metaman 當中。

提交與審核

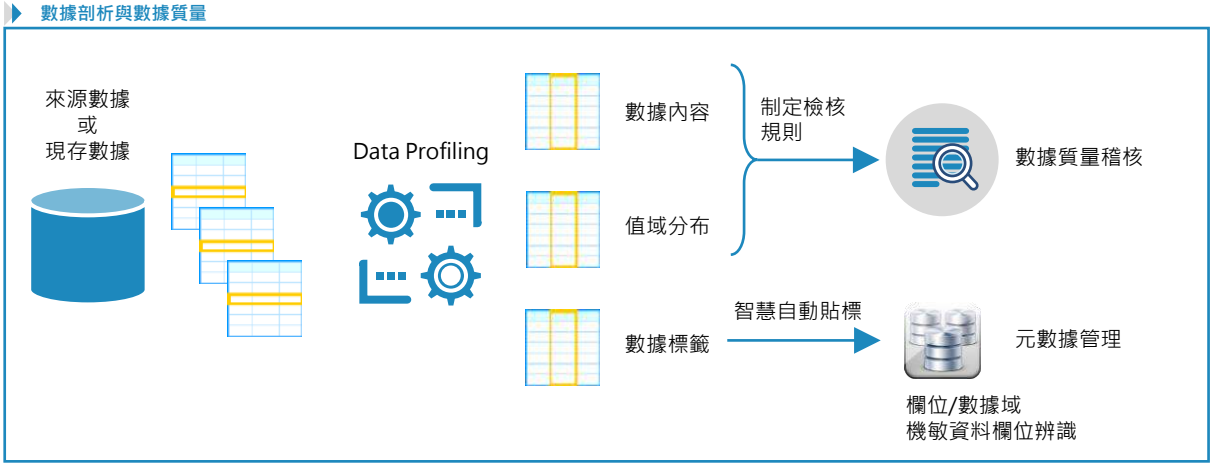
建立業務元數據與應用元數據時，會經過編輯、提交、審核及發布的流程，確保相關人員對發佈的資訊具有共識，才可發布至查詢平台提供企業中各部門人員查詢。技術元數據無須審核，但仍須經過具備相關權限的人員進行發布。

查詢

查詢平台簡單易用，透過瀏覽器即可操作使用。查詢一個項目時，系統會自動分析及比較元數據資訊，自動將該項目的詳細資料內容當中出現的相關名詞加上連結，點擊連結即可交互參照名詞之間的關係。



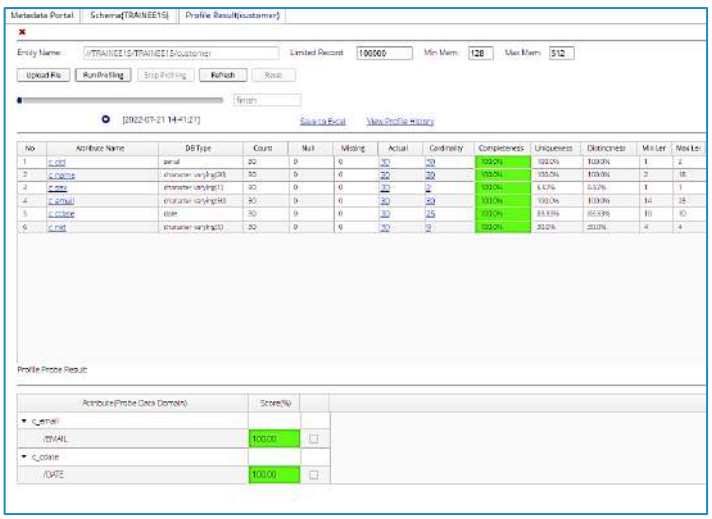
數據剖析與數據質量



數據剖析(Data Profile)

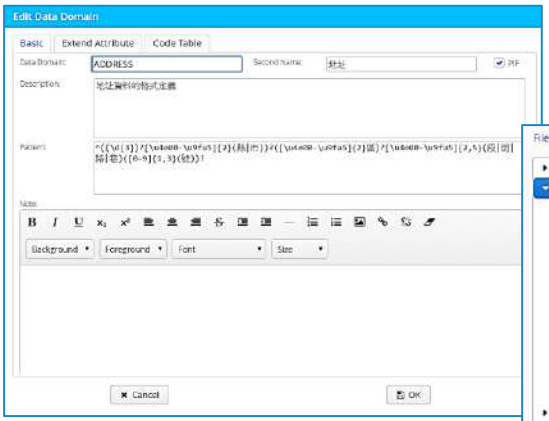
針對來源數據或現存數據(包含存在於資料庫或分析系統當中的資料)，經過數據剖析的過程，得到數據內容、值域分布及數據標籤。依據分析而得之標籤評分，可進行智慧自動貼標，數據剖析結果包含Top 10/Bottom 10 值域分析及圖形化剖析結果圖表。亦可訂定數據剖析自動排程，確保新進欄位與數據域的關係，快速識別特定的欄位或機敏資料相關欄位。

數據剖析與貼標結果



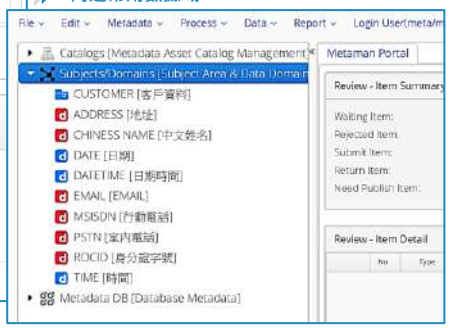
數據剖析的依據來自數據域(Data Domain)定義，用以定義數據的樣式、屬性、值域、格式等。除了內建常用的數據域，使用者可自訂數據域定義，亦可建立主題區(Subject Area)將數據域進行分類管理。

一旦執行過數據剖析，可透過數據域來查詢技術元數據中有哪些欄位符合此數據域的定義。



定義數據域

內建常用數據域



數據質量分析(Data Quality)

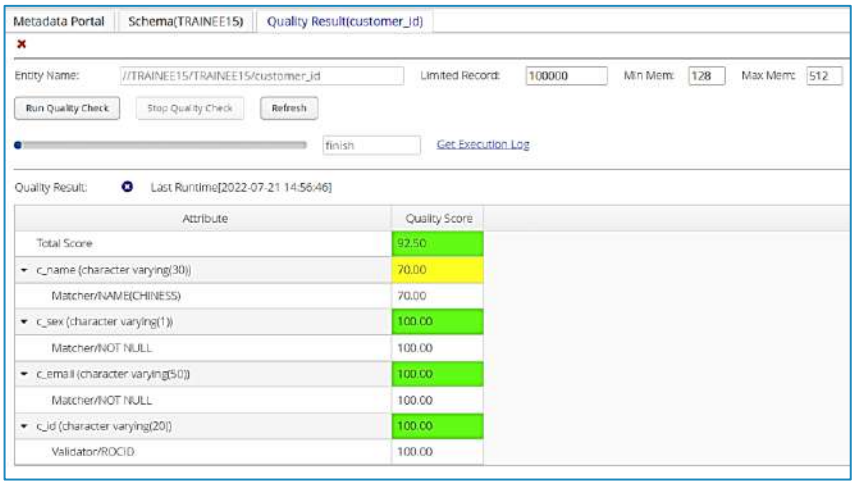
數據質量分析會依賴數據剖析後的貼標結果，來對數據域的定義制定分析規則。有時也可直接使用數據域中的定義資料進行檢核分析。系統內建多樣常見資料類型的檢核規則。此外也支援以Groovy腳本自定義質量檢核規則，方便依現實環境快速建置。

規則訂定完成後，再將需要執行數據質量檢核的欄位，選取適用的檢核分析質量規則，即可進行數據質量評分。可訂定數據質量分析自動排程，提供相關人員隨時掌握企業資料的情況。完成數據質量評分後，可運用於 Trinity ETL 作業設計之中，作為資料檢核過濾參照依據，將資料分別輸出，以供相關人員利用與查詢。

內建數據質量檢核規則

支援以Groovy腳本自定義規則

數據質量分介面



串流即時作業管理模組(SDM)

資料量的不斷膨脹，使各企業經常需要處理瞬息萬變大量湧進的資訊，資訊的即時處理成了一大難題，需求也日漸迫切。 Trinity串流即時作業管理模組以不間斷的持續作業執行模式，對多種不同的流數據類型提供近乎即時的處理能力。

Web化設計介面

Web 化的流數據作業設計介面，可輕鬆以圖型拖拉操作方式設計出流數據處理邏輯，節省複雜的程式編寫工作。

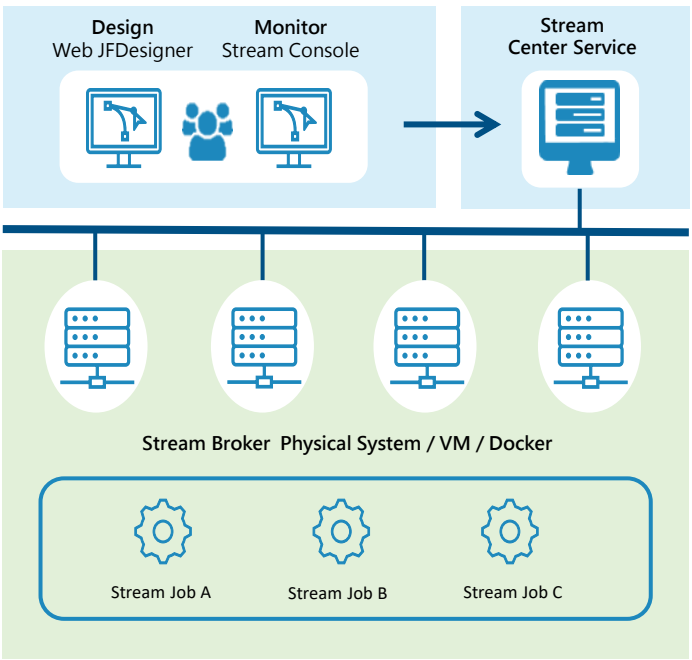
作業執行與監控

可運用串流即時作業發佈監控介面與串流即時作業中介模組執行與管理各作業。

支援多種流數據種類

- Redis DB 寫入
- Mongo DB 讀取與寫入
- 文字檔案的持續性寫出處理。
- 網頁爬蟲的持續性讀取處理。
- 日誌檔案內容的持續性監控讀取處理。
- 資料庫數據的持續性讀取與寫入處理。
- Restful Web Service 持續性呼叫處理。
- TCP/IP Socket 連線的持續性讀取與寫入處理。
- Message Queue(Kafka/Rabbit MQ/Active MQ/ IBM MQ) 的持續性讀取與寫入處理。

架構圖



Web Service 轉化組件(No code to Web Service)

無需撰寫程式既可將設計的資料處理邏輯轉化成 Web Service Restful API 呼叫介面提供其它系統進行呼叫，輕鬆整合進企業數據中台API架構。透過Broker的叢集擴充能力，可以微服務化方式進行管理對系統資源做有效分配。

分流/備援及可擴充性

提供流數據作業的獨立執行與管理機制，可橫向擴充Broker數量，使流數據作業分散於不同機器上執行，增加流數據作業的分流與備援執行機制。

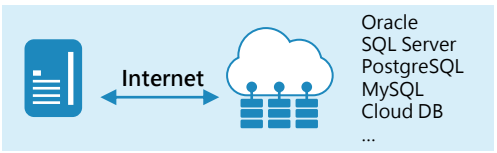
高應用性的數據處理組件

包含對流數據進行轉換/分流/合併的數據處理組件，可直接透過介面設定方式進行處理。

雲端服務

雲端服務興起，Trinity 可友善地支援雲端服務的數據處理交換。可架構使用雲端主機服務、雲端檔案服務，與雲端資料庫服務。

雲端服務常見情境



- 僅使用雲端服務上的資料庫服務
- 資料更新依然由地端主機來執行
- 使用者也會讀取/更新雲端資料庫

- 僅使用雲端服務上的主機服務或檔案服務
- 更新依然由地端主機來執行

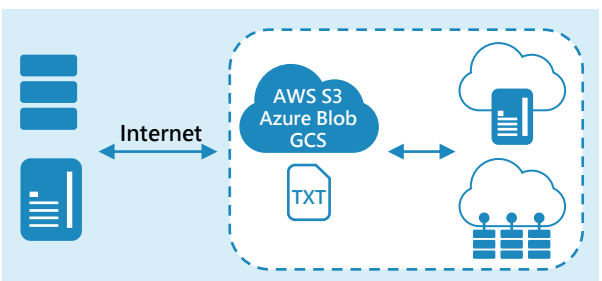
- 使用雲端服務上的主機服務 + 資料庫服務
- 資料交換發生在雲端上,僅少量資料需要到地端

- 使用雲端服務上的主機服務 + 資料庫服務
- 與地端資料庫或地端系統也會有大量資料交換作業

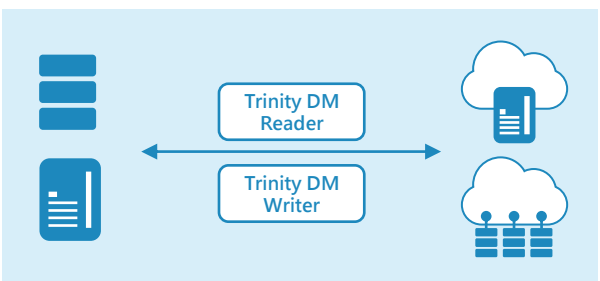
Trinity ETL Plugin 直接串接雲端資料庫與檔案服務

依據企業使用雲端服務的架構或目的的不同，ETL 處理器可布署於地端或雲端。而資料交換方式可為先落地再進行資料處理，或透過 Trinity 處理後不落地直接寫入。

Trinity 提供 AWS S3/Azure Blob/GCS 等檔案讀寫，支援 Text/CSV/XML/Json 格式檔案，直接串接雲端檔案服務，或使用 JDBC/API 介接資料庫服務。



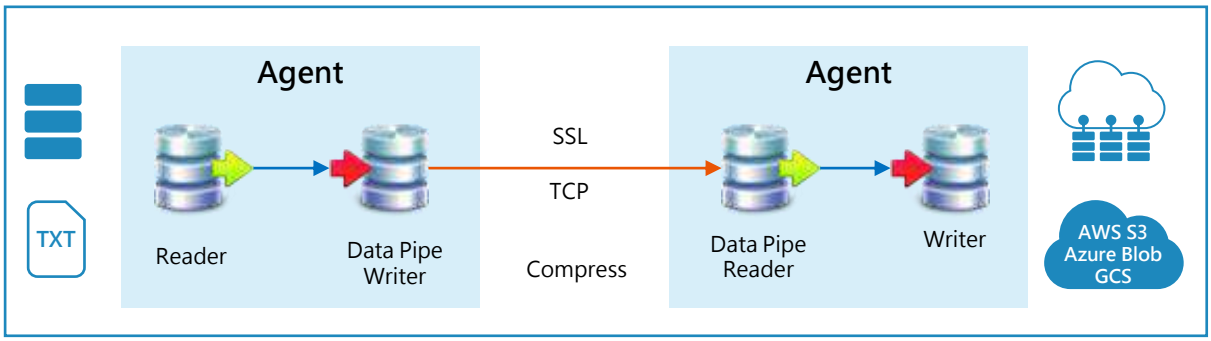
Trinity 支援先將資料傳至雲端檔案系統中，再由雲端主機將資料寫入至雲端資料庫或雲端系統中。(反之亦然)



Trinity 支援直接將資料從地端寫入至雲端資料庫或雲端系統中。(反之亦然)

Trinity 內建安全傳輸

Trinity 提供 Data Pipe Writer 與 Data Pipe Reader 用來在兩台機器間傳遞資料，包含地對地、地對雲，與雲對雲，皆可透過壓縮功能以減少網路傳輸使用頻寬。

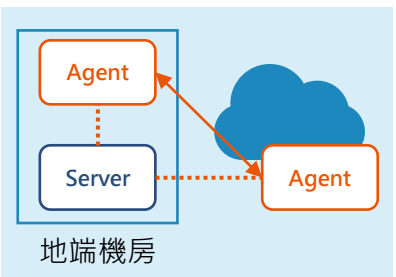


優點

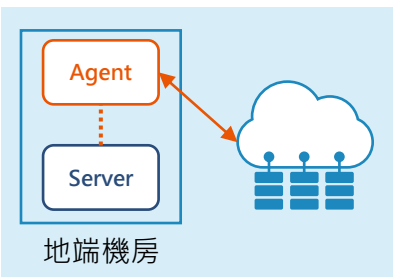
- 本地端對資料庫或文件讀取維持本地讀取的速度。
- Data Pipe Writer 會透過網路將資料以壓縮格式傳送至另一台 Trinity Agent 中的 Data Pipe Reader，以節省頻寬與加速。
- Data Pipe Reader 接收資料後可由後續的 Writer Plugin 寫入到本地端/雲端的資料庫或檔案系統。
- Data Pipe Writer/Data Pipe Reader 的資料格式定義與 ETL 作業一致，對 Trinity ETL 作業來看在設計面上並無不同，只需定義兩個不同作業使用 Flow 來同時執行。
- 不同端的 Trinity Agent 間只需讓防火牆開放 Data Pipe Plugin 使用的連線 port 即可。

Trinity 配置架構

如果所有系統是佈署在雲端，可將 Trinity 完整的 Trinity Server 與 Trinity Agent 也都安裝在雲端中，可使用 Data Pipe Plugin 安全傳輸。支援 JDBC/API 介接雲端資料庫與檔案服務，無額外的地端傳輸成本。



如果雲端服務僅是完整系統的一部分，Trinity Server 將會安裝在本地端，雲端僅安裝 Trinity Agent 來執行 ETL 作業，於兩端都有安裝 Trinity Agent 的架構下，可使用 Data Pipe Plugin 安全傳輸。支援 JDBC/API 介接雲端資料庫與檔案服務，有少量的地端傳輸成本。



如果雲端服務僅使用資料庫服務而沒有主機服務，且 Trinity Server 與 Trinity Agent 都將會安裝在本地端，則無法使用 Data Pipe Plugin 功能。可支援 JDBC/API 介接雲端資料庫與檔案服務，有大量的地端傳輸成本。

耐特普羅資訊股份有限公司
台北市內湖區民權東路六段23號七樓之三
電話：886-2-8791-0011 分機 788~789
傳真：886-2-8792-2700
網址：www.trinity-data.com/