
Table of Contents

Introduction	1.1
Baisc	1.2
Live-update Server	1.2.1
Project List	1.3
InQuire App	1.3.1
InQuire Web	1.3.2
Gerrit	1.3.3

Introduction

Version 0.1.1

本交接文檔使用 [Gitbook](#) 撰寫，並提供以下格式：

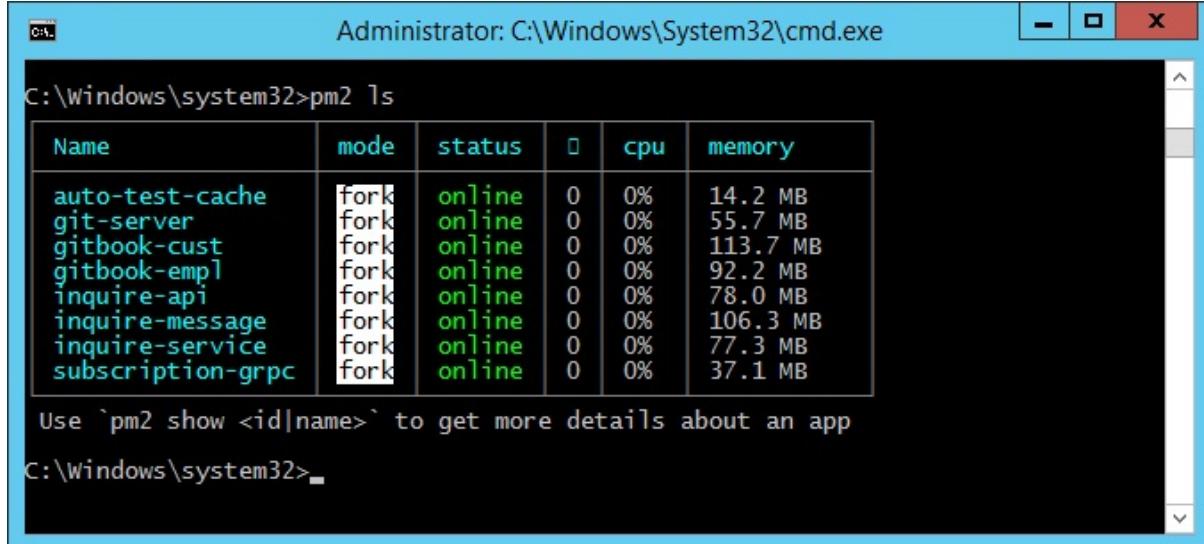
- GitBook: <http://inquire.insyde.com:24880/>
- PDF: [Click to Download](#)
- EPUB: [Click to Download](#)
- DOCX: [Click to Download](#)

Basic

目前僅描述 Live-update Server 上的設定參數，除了本文件外還另有一份手寫的 Live-update Server 上各個 Service 的 Administrator 帳號密碼

Live-update Server

Live-update 各個 Service 所使用的 Port 整理在 `C:\README.md`，目前有註冊啓動時會執行的 Script `c:\startup.ps1`，其內容是使用 [PM2](#) 去啓動管理各個 Service



```
Administrator: C:\Windows\System32\cmd.exe
C:\Windows\system32>pm2 ls
  Name      mode  status   □  cpu    memory
+---+-----+-----+-----+-----+
| auto-test-cache | fork | online | 0 | 0% | 14.2 MB
| git-server      | fork | online | 0 | 0% | 55.7 MB
| gitbook-cust    | fork | online | 0 | 0% | 113.7 MB
| gitbook-empl    | fork | online | 0 | 0% | 92.2 MB
| inquire-api     | fork | online | 0 | 0% | 78.0 MB
| inquire-message | fork | online | 0 | 0% | 106.3 MB
| inquire-service | fork | online | 0 | 0% | 77.3 MB
| subscription-grpc | fork | online | 0 | 0% | 37.1 MB
+---+-----+-----+-----+-----+
Use `pm2 show <id|name>` to get more details about an app
C:\Windows\system32>
```

Directory Structure

各個 Service 的 Binary 並不是放在 `C:\Program Files` 之下，而是 `C:\01.web_service` 目錄

- Service Binary
 - [Apache HTTP Server](#)
 - Path: `C:\01.web_service\Apache2.2`
 - Target: 原本使用 `80` Port 的 Web Server，目前棄用
 - [Gitea](#)
 - Path: `C:\01.web_service\gitea`
 - Target: 方便部署用的 Git Server
 - [MariaDB](#)
 - Path: `C:\01.web_service\MariaDB 10.1`
 - Target: 與 MySQL 相容，目前未使用
 - [Nginx](#)
 - Path: `C:\01.web_service\nginx-1.11.9`
 - Target: Web Server 兼 Reverse Proxy Server，使用 `80` 與 `8080` Port
 - [PostgreSQL](#)
 - Path: `C:\01.web_service\PostgreSQL9.6`
 - Target: 功能比較強大的 RDBMS，目前 `Subscription` 會拿來存儲已讀記錄與我的最愛
 - [RabbitMQ](#)
 - Path: `C:\01.web_service\RabbitMQ Server`
 - Target: Message Queue Server，原本是要給 InQuire 的 Notification Web API 使用的

我已經寫了一些基本的 Batch Script 已方便重啓 Service，亦放在 `C:\01.web_service` 下

- Nginx Restart Steps
 1. `C:\01.web_service\taskkill_nginx.bat`
 2. `C:\01.web_service\stop_nginx.bat`
 3. `C:\01.web_service\start_nginx.bat`

- **Directory Tree**

```
C:\  
+--01.web_service  
| +--Apache2.2  
| +--gitea  
| +--MariaDB 10.1  
| +--nginx-1.11.9  
| +--PostgreSQL9.6  
| +--RabbitMQ Server  
| \--_projects  
|   +--api.inquire  
|   +--auto-test-cache  
|   +--doc.inquire  
|   | +--cust  
|   | \--empl  
|   +--message.inquire  
|   +--server.inquire  
|   \--subscription-grpc  
+--_file_storage  
| +--FileCenter  
| +--gitea-repositories  
| +--LiveUpdate  
| \--_reverse  
\--_sql_data  
  +--LiveUpdate  
  \--_reverse
```

Project List

目前手頭上的 Project 都可以在 [InQuire Git Server](#) 上找到，而 Repository 的實際存放位置在 `Live-update Server` 上的 `C:_file_storage\gitea-repositories\`。每一個 Project 都會有 `README.md` 概述解說架構，如何編譯或執行，因此本文檔並不會詳述，只會略微帶過

InQuire App

Desktop Application

- Projects
 - [InQuire_WL / InQuire_WL](#)
- To-do
 - [] 升級 CEFPython 到 v66.0, 感謝 Lampix 願意資助
 - [] Linux 版本安裝腳本 (for Live-update)
 - [] 使用 Python3 與 wxPython4.0 重構

InQuire 是使用了 Python2 寫成的 Desktop Application, 其目的是在不改變公司現有系統的前提下, 整合所有服務成為唯一入口。大部分的 Dependencies 已經透過 [Pipenv](#) 管理, 需要手動安裝的部分請參考 README.md

構成 InQuire 主體有三大 Framework: [wxPython Classic](#)、[CEF Python](#) 以及 [Twisted](#), 分別負責 GUI、WebView 以及 Event Loop

本應用使用 [PyInstaller](#) 編譯, 在 Windows 下使用 [InnoSetup](#) 打包為 Installer。目前 Linux 版本僅於 [Ubuntu 16.04 LTS](#) 與 [Ubuntu 14.04 LTS](#) 測試。目前尚未撰寫安裝用的 Bash Script, 不過在 Ubuntu 上其實也可以使用 Python Script 來處理

System Tray

- Projects
 - [kevin.chen / InQuireTrayTest](#)
 - [kevin.chen / LiveUpdateTray](#)

開發語言為 Kotlin 與 Java, 目前公司最多人使用的 Linux Desktop 環境是 Ubuntu Desktop, Ubuntu 在 18.04 之前的 LTS 版本使用的是 Unity, 而 Unity 的 System Tray 與 Windows 相比少了部分功能, 而高層希望在 Linux 的行為要與在 Windows 一樣, 因此改使用 AWT 來畫出 System Tray。

編譯流程使用 [Gradle](#) 管理, 目錄架構就是一般標準的 Java 目錄架構。要注意的是 Ubuntu 14.04 LTS (Trusty Tahr) 的 default-jdk 是 Java 7, 因此不能直接使用 Lambda

Start Menu

- Projects
 - [InQuire_WL / inquire-react-start-menu](#)

使用 [Create React App](#) 創建的 Web App, 會透過 InQuire 的 Twisted Web Server 得到當前使用者可使用的 Service 列表並渲染在網頁上

InQuire Web

Web Application and Web API

- Projects
 - [InQuire_WL / inquire-subscription](#)
 - [InQuire_WL / inquire-server](#)
 - [InQuire_WL / inquire-api](#)
- To-do
 - [] `InQuire_WL / inquire-subscription` 可以列出當前使用者所有可見的 Publications 列表，無需切換 Channel

InQuire 所提供的服務絕大部分都與 Web 相關，當應用需求不是那麼適合使用 wxWidgets 來呈現的情況下，使用傳統的 Web Site 或 [Single-page Application](#) 在開發及維護上是比較方便的

這三個 Project 皆使用 [Node.js](#) 開發，在 Live-update Server 上使用 [nvm-windows](#) 來管理 Node.js 版本，使用 [PM2](#) 來管理各個 Process。這些 Service 在 Live-update Server 上使用的 Port 請參考 Live-update Server 的 `C:\README.md`，啓動腳本請參考 `C:\startuo.ps1`

`InQuire_WL / inquire-subscription` 為 Single-page Application，使用 [React Starter Kit](#) 作為 Boilerplate，其目的是將原本於 [e-Insyde](#) 的 Publication 加上 Group 與 Category 屬性

`InQuire_WL / inquire-server` 使用 [Express' application generator](#)，，使用 [Pug \(Jade\)](#) 作為 Template Engine，主要提供 File Center 與 Training (Youtube) 的頁面

`InQuire_WL / inquire-api` 與 `InQuire_WL / inquire-server` 同樣都使用 Express.js，主要提供 Email Verification 與 [skype for Business](#) 的 Web API

Web API

- Projects
 - [InQuire_WL / auto-test-cache](#)
 - [InQuire_WL / auto-test-grpc](#)
 - [InQuire_WL / subscription-grpc](#)
- To-do
 - [] 使用 `InQuire_WL / auto-test-grpc` 取代 `InQuire_WL / auto-test-cache`

這三個服務皆使用 Golang 撰寫，其目的在於 Service Owner 為提供 Web API 而不是提供 Database 權限直接讀取，而頻繁讀取可能造成 Database 效能低落，因此自行建造具 Cache 機制的 Web API Service 來避免 Client 直接讀取 Database

Dependencies 以透過 [govendor](#) 來進行管理，目錄架構參考了 [Standard Go Project Layout](#)。其中 `InQuire_WL / auto-test-cache` 是較早期的應急 Project，用於改善執行 MySQL Query 需要數十秒的問題，而之後的新 Project `InQuire_WL / auto-test-grpc` 其目的是希望 Client 不要直接讀取 Database，而是全部透過 Web API 來進行 Auto-test 的相關操作，但目前尚未完工

`InQuire_WL / subscription-grpc` 是針對 Notification 的資料 Cache，Subscription Message 以及 User 對應的 Channel 資料皆是存放於公司的 ERP Database，但因效能不佳問題，需在本地備份避免頻繁存取

User Guide

- Projects
 - [InQuire_WL / inquire-user-guide](#)
 - [InQuire_WL / inquire-gitbook-theme](#)

InQuire 的 User Guide 使用 Gitbook 撰寫，詳細文檔請參考 [GitBook Toolchain Documentation](#)，為了處理 URI 路徑問題因此對 Default Theme 做了一些修改

Gerrit

Web Portal

- Projects
 - [InQuire_WL / flask-git-repos](#)

Git Web Portal 是模仿公司內部 SVN Server 的既有網站行為，使用 [Flask](#) 所打造的入口網站。公司所導入的 Git Server 使用的是 [Gerrit Code Review](#)，由於 Gerrit 舊版本的 UI 界面不是太親民，而新版本的 UI 還在開發階段尚未完善且不一定會導入，因此透過一外部網站使用 [Gerrit REST API](#) 來取得 Project 列表與相關資訊，並加上額外的 Group 與 Owner 欄位來方便分類管理

- 既有的 SVN 入口網站



The screenshot shows a search interface for repositories. At the top, there's a header with the Subversion logo and the text "Insyde Subversion Entry of MingSheng headquarters". Below the header, there's a message: "Please use your domain account / password to navigate the repositories. The domain account will be 'domain_name\user_name'. Please use ALL LOWER CASE for the domain name and user name, or you may get a 'Forbidden' message." There are four input fields: "Filter", "Group Name" (set to "All Groups"), "Repository Name", and "Server Location" (set to "MingSheng headquarters"). Below these fields is a "Search" button. The main area is a table listing repositories:

Group	Repository	Path	Project Owner	Main	Location
Android	Froyo	http://svn.insydesw.com/Android/Froyo	Ray Peng		MingSheng headquarters
Android	Gingerbread	http://svn.insydesw.com/Android/Gingerbread	Jason Chen		MingSheng headquarters
Android	TEST0411	http://svn.insydesw.com/Android/TEST0411	Reno Chang		MingSheng headquarters
Android	TSET_Sync_1003	http://svn.insydesw.com/Android/TSET_Sync_1003	Reno Chang		MingSheng headquarters
Arcie	945G	http://svn.insydesw.com/Arcie/945G	SL Sheu		MingSheng headquarters
Arcie	Intel945GM_A68_945GM	http://svn.insydesw.com/Arcie/Intel945GM_A68_945GM	SL Sheu		MingSheng headquarters
Automation	H2O_Script	http://svn.insydesw.com/Automation/H2O_Script	Chiaming Hsiao		MingSheng headquarters
Bowser	Bowser	http://svn.insydesw.com/Bowser/Bowser	Stanley Ho		MingSheng headquarters
Chromium	Chromium_OS	http://svn.insydesw.com/Chromium/Chromium_OS	Jason Chen		MingSheng headquarters
CKT	Projects_ShareKanTan	http://svn.insydesw.com/CKT/Projects_ShareKanTan	Daniel WM Lin		MingSheng headquarters
Common-BIOS-Tools	BIOS-Tool-Library	http://svn.insydesw.com/Common-BIOS-Tools/BIOS-Tool-Library	Kanos Huang		MingSheng headquarters
Common-BIOS-Tools	CrossLib	http://svn.insydesw.com/Common-BIOS-Tools/CrossLib	Douglas Chen		MingSheng headquarters
Common-BIOS-Tools	CrossLib_Devts	http://svn.insydesw.com/Common-BIOS-Tools/CrossLib_Devts	Douglas Chen		MingSheng headquarters
Common-BIOS-Tools	H2OABT_W	http://svn.insydesw.com/Common-BIOS-Tools/H2OABT_W	David Yu		MingSheng headquarters
Common-BIOS-Tools	H2OATM_WL	http://svn.insydesw.com/Common-BIOS-Tools/H2OATM_WL	Chianning Hsiao		MingSheng headquarters
Common-BIOS-Tools	H2OELV_DWL	http://svn.insydesw.com/Common-BIOS-Tools/H2OELV_DWL	Tom Pan		MingSheng headquarters
Common-BIOS-Tools	H2OEEZ_WL	http://svn.insydesw.com/Common-BIOS-Tools/H2OEEZ_WL	Duncan Lin		MingSheng headquarters
Common-BIOS-Tools	H2OFFT_DWL	http://svn.insydesw.com/Common-BIOS-Tools/H2OFFT_DWL	Kanos Huang		MingSheng headquarters
Common-BIOS-Tools	H2OLRT_SD	http://svn.insydesw.com/Common-BIOS-Tools/H2OLRT_SD	Tom Pan		MingSheng headquarters
Common-BIOS-Tools	H2OQAE_DW	http://svn.insydesw.com/Common-BIOS-Tools/H2OQAE_DW	Alvin Chen		MingSheng headquarters
Common-BIOS-Tools	H2OPCM_WL	http://svn.insydesw.com/Common-BIOS-Tools/H2OPCM_WL	Chianning Hsiao		MingSheng headquarters
Common-BIOS-Tools	H2ORTE_SDWL	http://svn.insydesw.com/Common-BIOS-Tools/H2ORTE_SDWL	Alvin Chen		MingSheng headquarters
Common-BIOS-Tools	H2OSDE_DWL	http://svn.insydesw.com/Common-BIOS-Tools/H2OSDE_DWL	Alvin Chen		MingSheng headquarters
Common-BIOS-Tools	H2OUVE_SDWL	http://svn.insydesw.com/Common-BIOS-Tools/H2OUVE_SDWL	Kanos Huang		MingSheng headquarters
Common-BIOS-Tools	InQuire_WL	http://svn.insydesw.com/Common-BIOS-Tools/InQuire_WL	David Yu		MingSheng headquarters
Common-BIOS-Tools	LiveUpdate_WL	http://svn.insydesw.com/Common-BIOS-Tools/LiveUpdate_WL	David Yu		MingSheng headquarters

Plugin

- Projects
 - [kevin.chen / gitiles](#)

Gerrit 本身並沒有提供 Web UI 來顯示 Source Code，通常是透過 [GitWeb](#) 或是 [Gitiles](#) 來呈現。GitWeb 使用 Perl CGI 來實作，Gitiles 則是基於 [JGit](#)、[Eclipse Jetty](#) 與 [Closure Templates](#) 所撰寫的 Git Browser。目前的功能需求是將 [Google Repo Manifest](#) 檔案轉譯為目錄結構呈現，理論上是使用 Gitiles 來實現此功能會比較容易

Digression

最近在 GitHub 上看到一個有趣的 Project: [wmhilton/nde](#)，透過 JavaScript 將 Code `git clone` 至 Client 端顯示。但對於解析 Google Reo Manifest 這個需求還是在 Server 端處理渲染會比較好