

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace ForeignImport
{
    public partial class Form1 : Form
    {
        private CancellationTokenSource _cts;
        private static readonly object _logLock = new object();

        public Form1()
        {
            InitializeComponent();
            _cts = new CancellationTokenSource();
            StartBackgroundTask(_cts.Token);
        }

        private void StartBackgroundTask(Cancellation token)
        {
            Task.Run(() => SendQuery(token), token);
        }

        private void StopBackgroundTask()
        {
            _cts.Cancel();
        }

        protected override void OnFormClosing(FormClosingEventArgs e)
        {
            StopBackgroundTask();
            base.OnFormClosing(e);
        }

        private async Task SendQuery(Cancellation token)
        {
            List<string> lst = System.Configuration.ConfigurationManager.AppSettings["enable"]
                .Split(',')
                .Where(x => new string[] { "fi001t", "fi004t", "fi006t" }.Contains(x))
                .ToList();

            while (!token.IsCancellationRequested)
            {
                try

```

```

{
    if (DateTime.Now.Hour == cleanhr && DateTime.Now.Minute % 1 == 0)
    {
        foreach (string name in lst)
        {
            CleanAll(name);
        }
    }

    Logger("搜尋待發查資料...");
    foreach (string name in lst)
    {
        var data = getdata(name);
        await QueryAll(data);
    }

    await Task.Delay(scangap, token); // 非阻塞的延遲
}
catch (OperationCanceledException)
{
    Logger("背景任務已取消");
    break;
}
catch (Exception ex)
{
    Logger($"發生錯誤: {ex.Message}");
}
}
}

private async Task QueryAll(List<QueryRow> qrlist)
{
    foreach (var qr in qrlist)
    {
        try
        {
            var req = new Request(qr);
            string result = mode == "0" ? req.SendRequest() : GetTestString(qr.tablename);
            var data = Parsedata(result);

            if (data.rcode == "0000")
            {
                req.SaveData(data);
                Logger($"{qr.tablename} 序號[{getSN(qr)}] 發查成功, 已新增資料");
            }
            else
            {

```

```

        Logger($"{qr.tablename} 序號[{getSN(qr)}] 發查錯誤! 錯誤代碼:
{data.rcode}");
    }
}
catch (Exception ex)
{
    Logger($"查詢發生錯誤: {ex.Message}");
}
}
}

private void Logger(string content)
{
    lock (_logLock)
    {
        content = $"{DateTime.Now:yyyy/MM/dd HH:mm:ss} {content}\r\n";
        File.AppendAllText(logpath + "/" + DateTime.Now.ToString("yyyyMMdd") + ".log",
content, Encoding.Default);
    }
}

private void CleanAll(string tablename)
{
    string yesterday = DateTime.Now.ToString("yyyy-MM-dd");
    string qry = "";
    string deltable = "";

    switch (tablename.ToUpper())
    {
        case "FI001T":
            deltable = "fi002t";
            qry = $"delete from {schema}fi002t where insert_time<'{yesterday}'; ";
            break;
        case "FI004T":
            deltable = "fi005t";
            qry = $"delete from {schema}fi005t where insert_time<'{yesterday}'; ";
            break;
        case "FI006T":
            deltable = "fi006t";
            qry = $"update {schema}fi006t set result=" where insert_time<'{yesterday}'; ";
            break;
    }

    DaoDbCmd.query(qry);
    Logger($"刪除昨日 {deltable} 的資料");
}
}
}
}

```