

醫療資料傳輸共通介面 API 安裝及使用說明(UI版)

修訂日期：106年12月15日

備註：若有使用疑問請洽健保資訊網服務系統
首頁下方資訊技術諮詢服務窗口

目 次

壹、	執行環境需求	3
貳、	改版紀錄.....	4
參、	作業方式說明	10
肆、	作業機制.....	12
伍、	安裝方式.....	13
陸、	使用說明.....	18
柒、	常見問題.....	28
捌、	函式庫說明.....	29
玖、	API 使用範例	37
一、	Html 網頁.....	37
二、	VB 使用物件引用方式.....	38
三、	VB 使用 CreateObject 的呼叫方式.....	39
四、	VC++ 使用 CoCreateInstance 的呼叫方式.....	40
五、	VC++ 使用 Invoke 的呼叫方式	41
壹拾、	附件.....	44
一、	NHI_EII_DLL_i.c.....	44
二、	NHI_EII_DLL.h.....	46

壹、執行環境需求

- (一) Microsoft Windows 平台。
- (二) 已安裝醫療資料傳輸共通介面應用程式的個人電腦。
- (三) 已連接健保 IC 卡之讀卡機。
- (四) 網路環境需開放連線至 10.253.253.250 及 10.253.253.241 服務為 tcp 12100,12200,12300,12400,12500,12600。
- (五) 為確保系統運作正常，請先自行升級 IE 為 6.0 以上版本及安裝最新的 ADO 元件（均可於微軟網站免費下載）。
- (六) 傳輸過程相當耗用系統資源，若傳輸的檔案較大時，建議使用等級較高的電腦進行作業。（一般而言，電腦可用記憶體至少需要檔案大小的二倍，例如要傳輸 500MB 的檔案，至少要 1GB 的可用記憶體）。

貳、 改版紀錄

(一) 95.07.12

1. 首次公開測試版本。

(二) 95.07.24

1. 加強檢查需輸入的參數。
2. 加強檢查需輸入的參數。
3. 自動修正當忽略首次輸入參數時產生的問題。
4. 支援 Win 95/98/2000/XP 作業系統。但 Win95 因本身架構穩定性及元件支援較少，建議儘可能使用 Win98 以上版本。
5. 取消在產生的狀態紀錄。
6. 修正當多次上傳失敗，重傳時會有造成的 Memory leak 現象。

(三) 95.07.31

1. 修正前版遺漏的附加元件 MSDAODC.OCX。
2. 修正重複啟動應用程式的檢查。
3. 增加對 Reader 作業的逾時控制。
4. 加強作業環境是否就緒的檢查。
5. 修正作業畫面狀態的顯示。
6. 修正當同一檔案嘗試上傳作業失敗超過 5 次，先將其擱置。
7. 增加對當檔案放置尚未就緒時產生異常的異常狀況控制。
8. 增加異常訊息的顯示。

9. 修正為將原件放置改放於應用程式所在位置，因 Windows 作業系統當元件放置於系統目錄時，元件註冊作業可能會無效。
10. 新增「共通傳輸作業平台使用說明.doc」。

(四) 95.08.31

1. 正式版本 1.5 版。
2. 新增版本自動更新功能，未來每次啟動連線時會自動檢查有無新版，並可讓使用者選擇是否更新。
3. 新增傳輸異常中止時下次自動續傳功能。
4. 修正一些之前版本的 bugs。
5. 所有文件整合成「共通傳輸平台安裝及使用說明.doc」。

(五) 96.04.10

1. 正式版本 1.6 版。
2. 加強傳輸異常中止時下次自動續傳功能。
3. 增加代碼檔更新機制。
4. 修正一些之前版本的 bugs。

(六) 96.05.23

1. 正式版本 1.6a 版。
2. 修正:連線時出現錯誤訊息「元件或它的依存檔案並未正確的註冊:某檔案遺漏或不正確」。

(七) 96.08.17

1. 正式版本 2.0 版。

2. 放大可接收單一檔案大小為 500MBytes。
3. 加強傳輸功能及新增錯誤訊息。

(八) 98.06.04

1. 正式版本 2.1 版。
2. 加強傳輸有效性檢查功能。
3. 本版起將不再支援 Win95/98，Win95/98 的使用者請安裝 2.0 版。

(九) 99.08.25

1. 正式版本 2.2 版。
2. 加強傳輸穩定性。
3. 增加讀卡機參數設定及紀錄維護工具程式。
4. 新增 NHI_SENDA 函式。

(十) 101.06.18

1. 正式版本 2.3 版。
2. 新增 NHI_Download 函式。
3. 新增 NHI_Get 函式。
4. 增訂異常代碼。
5. 網路連線新增 12500,12600 兩個 port。

(十一) 101.7.17

1. 本文件格式調整。
2. 修訂前次版本安裝異常。
3. 原先提供版本具備 UI 介面（簡稱 UI 版），使用者可由

畫面得知本網站相關作業處理結果，101 年 7 月 16 日新增「簡易版」則僅供醫事機構資訊系統撰寫程式使用，不具備 UI 介面，以簡化醫事機構資訊系統撰寫程式之程序；若是上傳電子化醫療影像檔案作業，須使用 UI 版，以處理檔案續傳作業。

4. 新增含微軟 ADO 元件之安裝檔，於首頁左側下載專區之檔案說明為「醫療資料傳輸共通介面 API 元件(UI 版，含 MS ADO 元件安裝)」。

(十二) 101.11.22

上傳作業種類增加' 06' 抽樣回饋資料。

(十三) 102.7.23

配合組織改制，修改名稱為「衛生福利部中央健康保險署」。

(十四) 103.4.24

檔案大小限制放大可接收單一檔案大小為 1GB。

(十五) 103.7.28

加強傳輸功能，提升資料傳輸速度。

(十六) 103.9.19

修正 NHI_Download 函式異常問題。

(十七) 104.2.16

上傳作業種類增加' 07' 醫療費用電子申復資料及' 09' 預檢醫療費用電子申復資料。

(十八) 105.12.19

1. 正式版本 3.0 版。
2. 檔案大小限制放大可傳送單一檔案大小為 4.5GB。

3. 檔案收送改由 windows 背景服務程式執行。
4. 上傳作業種類增加'14'事前審查 XML 格式批次上傳,'15'費用抽審 XML 格式批次上傳,限 3.0 以上版本使用。

(十九) 106.01.17

3.0 版支援之作業系統:Windows7(64 位元)、Windows8.1(64 位元)、Windows10(32/64 位元)。

(二十) 106.03.17

3.1 版可自動線上下載及更新程式。

(二十一) 106.04.14

安裝使用 3.0 (含)以上版本之 NHI_EII 服務,請不可同時執行 2.3 版的 NHI_EII_Main.exe。

(二十二) 106.05.17

1. 正式版本 3.2 版。
2. 上傳作業種類增加'25'牙位更正 XML 檔案。
3. 下傳作業種類增加'04'電子轉診資料下載_回復電子轉診單
4. 支援網路磁碟機檔案傳送。
5. 修訂前次版本無法自動更新版本之異常

(二十三) 106.06.27

1. 上傳作業種類增加'26'檢驗(查)每日上傳資料 XML 格式
2. 下傳作業種類'26'檢驗(查)每日上傳資料 XML 格式檢核結果回饋資料。

(二十四) 106.07.04

上傳作業種類增加' 16'重大傷病 XML 格式批次上傳。

(二十五) 106.12.15

1. 正式版本 5.0 版。

2. 上傳作業種類增加' 90' CT/MRI 上傳文字檔案，'
91' CT/MRI 上傳影像檔案。

3. 下傳作業種類' 89' CT/MRI 檢核結果回饋資料。

(二十六) 107.1.6

修正 API 使用範例

參、作業方式說明

醫療資料傳輸共通介面應用程式共包含 8 個主要檔案：

- (一) NHIEIService.exe 功能：於背景執行傳輸作業執行流程，開啟關閉請用 Windows 服務或 NHI_EII_Main.exe。
- (二) NHI_EII_View.exe 功能：系統環境設定維護介面，當 Reader.dll 變更路徑或當讀卡機連接的 COM Port 變更。監看設定控制檔案傳輸平台，開關傳輸服務，傳輸連通偵測、讀卡機安全模組偵測。
- (三) NHI_EII_C.dll：檔案傳輸平台用動態聯結函式，提供函式供檔案傳輸平台呼叫。
- (四) NHI_EII_D.dll：執行版本更新及應用程式版本置換。
- (五) NHI_EII_DLL.dll：醫療資料傳輸共通介面 API，提供給醫療資料系統呼叫作為醫療資料傳輸共通介面應用程式的介面。
- (六) mNHIICCDLL.exe：多元瀏覽器背景服務程式，如果需要使用非 COM 元件多元瀏覽器網頁服務就必須使用本服務。開啟關閉請用 Windows 服務或 mNHIApp.exe。
- (七) NHIEIIDLG.exe：多元瀏覽器檔案選取視窗程式。開啟關閉請用 Windows 服務或 mNHIApp.exe。
- (八) mNHIApp.exe：多元瀏覽器背景服務程式開啟關閉，本程式需要管理者權限才能開啟關閉，否則只能用於偵測多元瀏覽器服務是否正常。
- (九) NHIEIIUpdate.exe：共同傳輸平台 Client 端線上下載更新程式，本程式需要管理者權限。安裝後自動放置於啟動目錄，如果發現程式需要更新，會自動下載至暫存目錄後詢問使

用者是否安裝。

肆、作業機制

- (一) 醫療資料傳輸共通介面是透過現有醫療院所的健保讀卡機設備，將醫療院所欲上傳至 IDC 的資料先執行認證後執行簽章完成後執行加密，將資料的密文上傳至 IDC，以確保傳輸資料的安全及可信任性。
- (二) 透過讀卡機安全模組與 IDC 的認證可確認傳遞者的身分，確實為醫療院所本身。
- (三) 傳輸的資料透過讀卡機安全模組的簽章可達到資料的完整「無法被竄改」及不可否認性。
- (四) 傳輸過程資料的加密可達到資料傳輸過程的隱密性。
- (五) 使用讀卡機的目的是為了讓傳輸的資料的過程達到「認證」、「完整無法竄改」、「不可否認性」及「隱密」的機制以維護資訊安全。
- (六) 本系統自 3.1 版起已經提供版本自動更新及續傳的功能。

伍、安裝方式

(一) 健保資訊網服務系統 (VPN) 入口網站

https://medvpn.nhi.gov.tw，點選下載專區。

The screenshot shows the homepage of the National Health Insurance Administration's VPN service. The header includes the NHI logo and the text '衛生福利部中央健康保險署 健保資訊網服務系統(VPN)'. A maintenance notice states: '本系統於每日 18:00~19:00、1:00~1:30 例行維護，請儘量勿於該時段使用系統，謝謝您的配合。' The main navigation area has a 'Download Area' (下載專區) button highlighted with a red box. Below it, a table lists various files for download, including VPN introduction documents, preparation guides, and installation files. The right sidebar contains a 'Service Login' (服務登入) section with options for different user types like medical staff and general users.

檔案說明	檔案	更新日期
本網站簡介	VPNIntro10201.pdf VPNIntro10201.ppt	102/02/01
入口網站介紹	VPN_WebSite.pdf VPN_WebSite.doc	101/09/22
使用本網站之院所準備事項說明	VPN_Prepare.pdf VPN_Prepare.doc	102/02/01
[適用健保讀卡機]電子憑證元件安裝說明	VPN_ECInstall.pdf VPN_ECInstall.doc	101/04/25
[適用健保讀卡機]健保資訊網憑證元件(適WinXP以上)	IDC_Setup.msi.zip	102/06/06
[適用健保讀卡機]健保資訊網憑證元件(適Win98或2000)	IDC_Setup.exe.zip	101/06/01
健保讀卡機之驅動程式(東元:reader.dll.1.zip;瑞茂:reader.dll.2.zip)	Reader.dll.1.zip Reader.dll.2.zip	101/04/07

(二) 服務項目選取「共通作業」，按「查詢」。

This screenshot shows the same website as above, but with the 'Service Items' (服務項目) dropdown menu set to 'Common Work' (共通作業), which is highlighted with a red box. The 'Query' (查詢) button is also visible next to the dropdown. The rest of the page layout, including the maintenance notice and the file list, remains the same.

(三) 在下面圖示框選處可以看到下載檔案。

檔案說明	檔案	更新日期
醫事機構憑證IC卡申辦流程	HCA_HOSP.pdf	102/07/23
醫事人員憑證IC卡申辦流程	HCA_PRSN.pdf	102/07/23
常見問題說明	VPN_QA.pdf CA.zip VPN_QA.doc	105/01/04
ICD-9-CM手術及處置代碼對應檔	ICD-9-CM.xls	104/04/20
[適用健保讀卡機]健保資訊網憑證元件檢測	IDCVer_CHK.hta	104/06/05
[適用健保讀卡機]健保資訊網憑證元件檢測說明	IDCVer_CHK.doc	104/06/05
醫療資料傳輸共通介面 API使用及安裝說明(簡易版)	NHI_EII_S.doc NHI_EII_S.pdf	105/12/20
醫療資料傳輸共通介面 API元件(簡易版)	NHI_EII_S_d1051219.zip	105/12/20
醫療資料傳輸共通介面 API元件(UI版)	NHI_EII_UI_1051219.zip	105/12/20

(四) 選取下載「醫療資料傳輸共通介面 API 元件(UI版)」。

(五) 如果安裝舊版共通傳輸平台請移除

(六) 下載完成後執行 Setup.exe 出現畫面





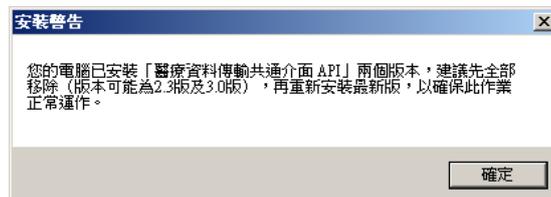
(七) 選取欲安裝的目錄



(八) 完成後按下一步進行安裝



(九) 已安裝舊版(2.3 版或 3.0 版)，會發出警告訊息建議先移除，
但不會影響安裝。



(十) 安裝完成



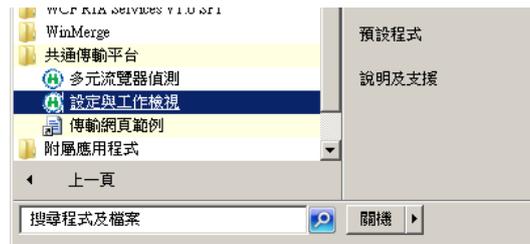
陸、使用說明

(一) 執行醫療資料傳輸共通介面應用程式

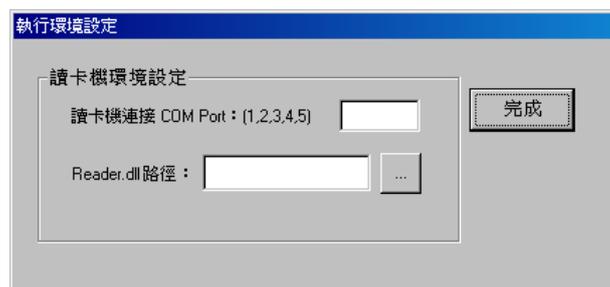
開始功能表→程式集→共通傳輸平台→設定工作與檢視。

(或 安裝目錄點選 NHI_EII_View.exe。

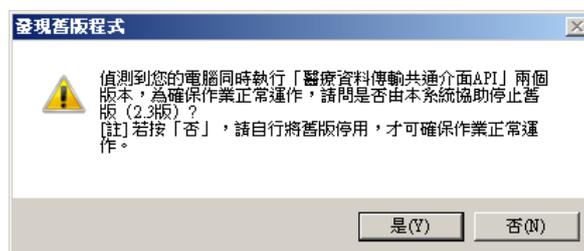
請注意，必須有管理者權限)。



(二) 首次執行時應用系統會先要求需輸入 Reader 連接的通訊連接埠及 Reader.dll 所在的位置。



(三) 同時執行兩個以上版本，會發出警告訊息建議先停止舊版。

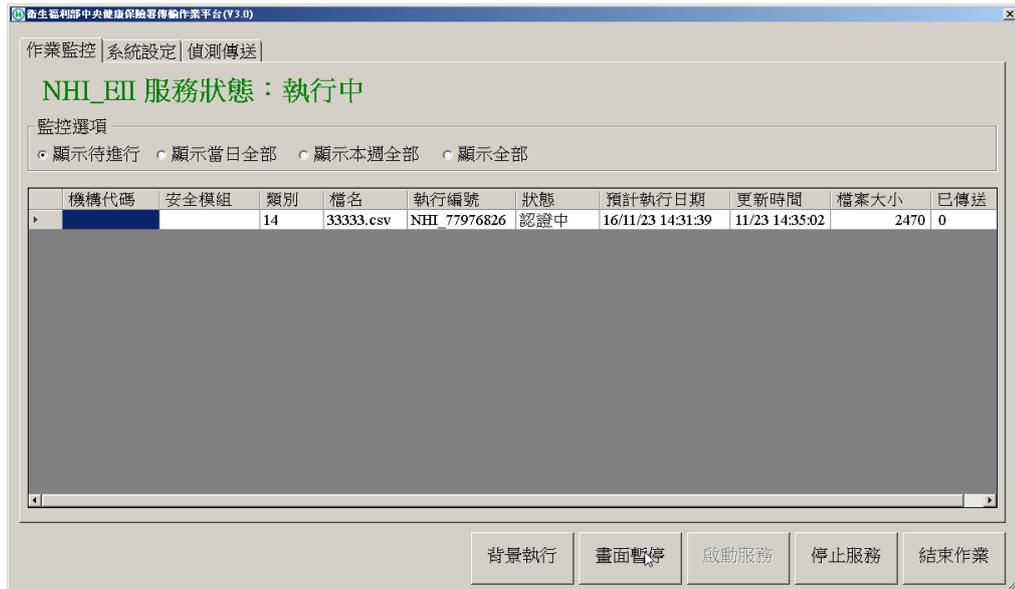


(四) 畫面說明

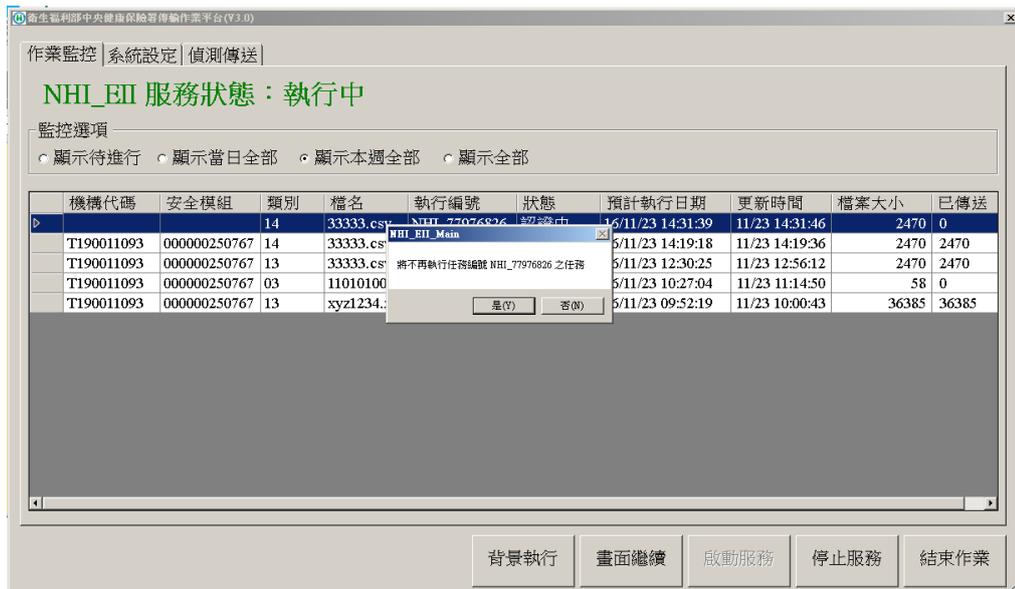


作業監控：

1. 點選「顯示待進行」顯示尚未完成的所有任務。
 2. 點選「顯示當日全部」顯示今日加入的所有任務。
 3. 點選「顯示本週全部」顯示最近七日加入的所有任務。
 4. 點選「顯示全部」顯示全部的任務。
 5. 「執行編號」為醫療資料傳輸共通介面的任務代號，可使用此代號至健保資訊網服務系統查詢作業狀態及作業時間。
 6. 「狀態」為該筆任務執行的狀態目前已定義狀態如狀態說明表。
 7. 背景執行應用程式將在隱藏置工具列內。
- (1) 畫面暫停：執行過程中畫面每秒會進行更新，當欲進行任務刪除或指定任務重新執行時點選畫面暫停方便編輯，在進行任務刪除時可以先點選畫面暫停，再行點選欲刪除的任務，該任務會呈現反黑，欲選取多筆請按下「Ctrl」鍵再點選欲刪除的任務即可。



(2) 按下「Del」鍵即會出現提示進行任務刪除。

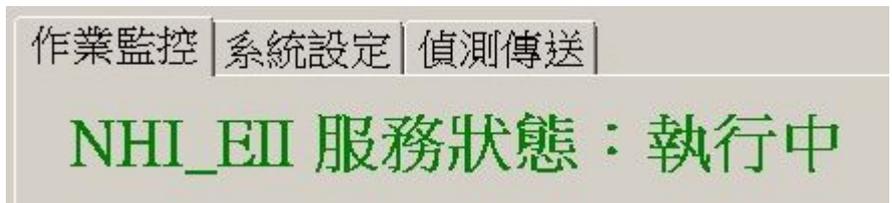


按下「是」即放棄該任務的執行。按下「否」即可繼續執行該任務。點選錯誤時可按下「Esc」鍵，回復執行狀態。

8. 啟動服務及停止服務

- (1) 左上角可以看到 NHI_EII 服務狀態，如果目前為綠色字執行中，表示目前背景傳送程式正在執行

中，可以按下右下方**停止服務**按鈕停止共通傳輸平台收送檔案服務



- (2) 如果為紅色字停止中，表示現在不會進行共通傳輸平台檔案收送，可以按下右下方**啟動服務**按鈕進行收送



9. 於 2.0 版起本系統於監控畫面上增加已傳送 Byte 數之資訊。

(五) 狀態說明表

1. 定義為「自動重試」程式會自動重試。

2. 定義為「不處理」程式將不再處理該任務。

狀態	說明	處理方式
待處理	代表該任務已排入傳送清單內等待處理	待處理中
認證中	代表該任務目前已進行向 IDC 取得傳輸授權碼	已處理中
認證失敗	向 IDC 取得傳輸授權碼失敗	自動重試
認證失敗 1	認證步驟 4 失敗，與中心端交互認證失敗	自動重試
認證失敗 2	連接 IDC 失敗，可能是網路不通	自動重試
認證失敗 3	內部認證失敗	自動重試
認證失敗 4	認證步驟 2 失敗	自動重試
認證失敗 5	外部認證失敗	自動重試
認證完成	代表該任務目前已取得傳輸授權碼	已處理中
檔案加簽	代表該任務目前正為檔案進行簽章	已處理中
簽章失敗	代表簽章作業失敗	自動重試
檔案無法存取	代表欲上傳的檔案有問題，可能無法開啟或檔案不存在。	不處理
檔案大小異常	代表該任務的檔案有問題，檔案大小為 0	不處理
傳送中	代表該任務正將檔案傳送至 IDC 中	已處理中
傳送失敗 1	代表該任務傳送作業不正常	自動重試
傳送失敗 2	代表該任務連接至 IDC 異常	自動重試
傳送失敗 3	代表該任務被 IDC 踢退可能是 1. 傳輸作業逾時。 2. 檔案超過可接收的範圍。	自動重試
重試次數過多	代表該任務已失敗超過 5 次，將先擱置待後續任務處理完後再予重試	自動重試
資料庫寫入異常	代表該任務作業紀錄異常	自動重試
通訊異常	代表網路無法連接至 IDC	自動重試
連線異常	代表網路無法連接至 IDC	自動重試

狀態	說明	處理方式
上傳完成等待處理中	代表資料已上傳至 IDC，正排程處理，請稍候	自動重試
已受理等待處理中	代表資料已上傳至 IDC，排程作業處理，請稍候	自動重試
檔案建立失敗，請檢查	請檢查硬碟空間或檔案存放位置是否有問題	自動重試
檔案寫入異常，請檢查	請檢查硬碟空間或檔案存放位置是否有問題	自動重試
檔案接收失敗	請檢查硬碟空間或檔案存放位置是否有問題	自動重試
已完成	代表該任務已正常上傳至 IDC	處理完成
失敗	不明原因異常	自動重試
放棄執行	使用者放棄該任務	不處理

(六) 作業異常訊息說明

訊息 1 至 4 為程式可繼續運作

1. 輸入讀卡機使用的 RS232 連接埠。



2. 需選取讀卡機使用的 Reader.dll 檔名需為 Reader.dll。



3. 所選取的檔案檔名不是 Reader.dll，請重新選取。



4. 版本更新查詢作業失敗，將在下一次啟動時再進行版本更新作業。



訊息 5 至 11 程式已無法繼續運作，所輸入的任務仍會保持在作業名單內，待環境允許後繼續傳送。

5. 無法載入，系統網路元件。



6. 讀卡機沒有回應，或作業逾時，設定做業逾時為 60 秒，若遇到這類狀況請重置讀卡機。



7. 請檢查是否已插入安全模組。



8. 請重開讀卡機。



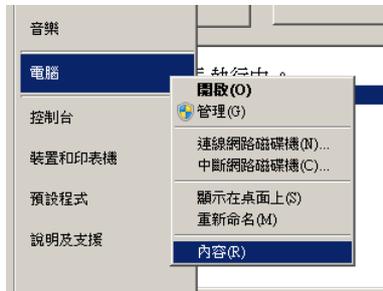
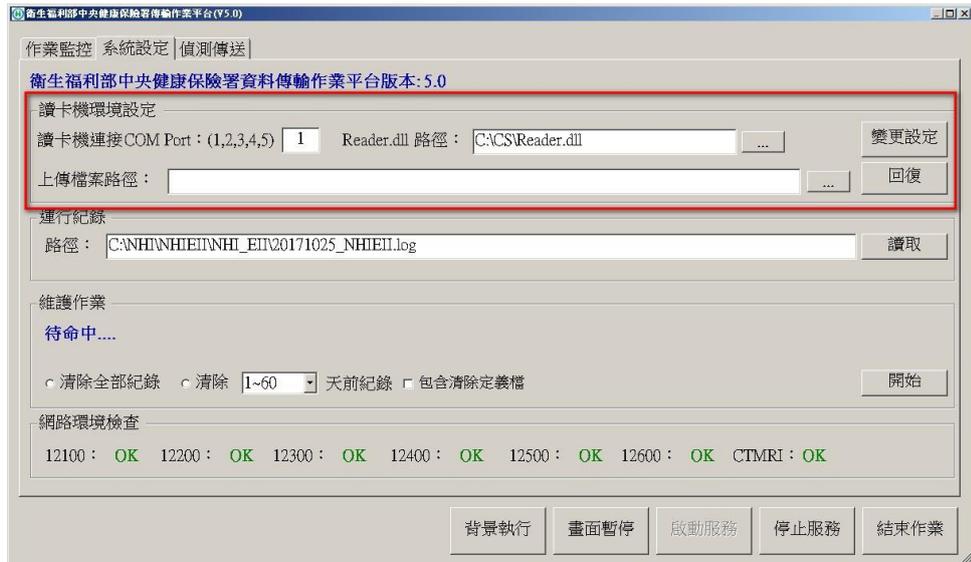
9. 讀卡機讀取作業異常。



10. 載入 Reader.dll 異常，可能是 Reader.dll 不是讀卡機廠商提供的 DLL。

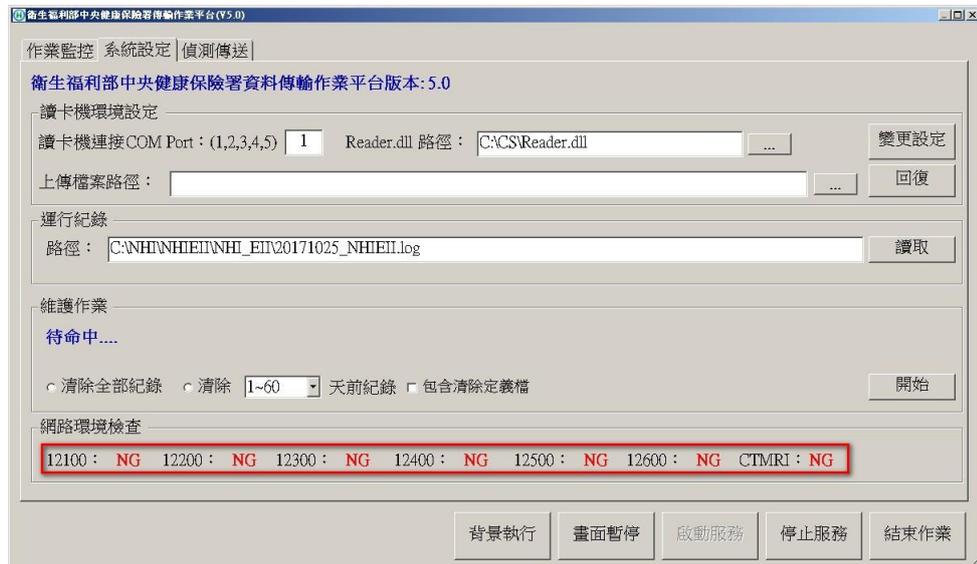


11. 可能是系統環境設定不正確，可選取 系統設定→讀卡機環境設定，確認 Reader.dll 是不是讀卡機廠商提供的 DLL，或是此時讀卡機已被別的作業佔據無法使用。



12. 系統設定畫面下方：網路環境檢查

正常狀態為綠色字體 OK，如果為紅色字 NG，請檢察
傳輸主機網卡 ip 設定至網路線....貴單位網路防火牆
port 號 12100~12600 至健保署主機之防火牆與路由狀
況是否暢通？



柒、常見問題

(一) 安裝過程失敗，可能原因如下：

1. 若系統使用者名稱為中文，請更改使用者名稱為英文。

2. 曾安裝過以前的版本，作法如下：

尋找系統內的檔案 NHI_EII_C.dll ，
NHI_PACS_DLL.dll，將其刪除；或尋找系統內 NHI_EII
的目錄，將其刪除後，應可正常作業。

(二) 發生重試次數過多

可能原因為檔案大小超過可接收範圍，請刪除該任務。

(三) 安裝使用 3.0(含)以上版本之 NHI_EII 服務，請不可同時執行 2.3 版的 NHI_EII_Main.exe。

捌、 函式庫說明

(一) 傳送 PACS 檔案

BSTR NHI_Send(BSTR sSourcePath, BSTR sDestName);

1. 函式功能說明：

- (1) 本函式將使用者端之檔案依指定時間、指定之安全機制傳送至 IDC。
- (2) 本函式每次傳送一個檔案，並將自動擷取醫療院所代碼及上傳作業日期時間。
- (3) 本函式作業若失敗，將回傳錯誤代碼。作業若成功，則傳回檔案執行編號，應用程式可依此代碼查詢傳送作業情形。

2. 參數說明：(*) 為必填

- (1) * sSourcePath:原始檔案目錄及名稱
- (2) * sDestName:目的檔案名稱

3. 回覆代碼：

E01: Data/File Not Found Error

E11: XML File Error

E12: Copy File Error

E13: Log DB Error

E98: Function Not implement

E99: Unknown Error

(二) 傳送醫療資料檔案

BSTR NHI_SendA(BSTR sSourcePath, BSTR sDestName,
BSTR sTypeCode);

1. 函式功能說明：

- (1) 本函式將使用者端之檔案依指定時間、指定之安全機制傳送至 IDC。
- (2) 本函式每次傳送一個檔案，並將自動擷取醫療院所代碼及上傳作業日期時間。
- (3) 本函式作業若失敗，將回傳錯誤代碼。作業若成功，則傳回檔案執行編號，應用程式可依此代碼查詢傳送作業情形。

2. 參數說明：(*) 為必填

- (1) * sSourcePath:原始檔案目錄及名稱
- (2) * sDestName:目的檔案名稱
- (3) sTypeCode：預設為 00
00：電子化專業審查系統使用
03：醫費申報資料 XML 格式
05：預檢醫費申報資料 XML 格式
07：醫療費用電子申復資料
09：預檢醫療費用電子申復資料
14：事前審查 XML 格式批次上傳
15：費用抽審 XML 格式批次上傳
16：重大傷病 XML 格式批次上傳
25：牙位更正 XML 格式批次上傳
26：檢驗(查)每日上傳資料 XML 格式
90：CT/MRI 上傳文字檔案
91：CT/MRI 上傳影像檔案

3. 回覆代碼：

E01: Data/File Not Found Error

E11: XML File Error
E12: Copy File Error
E13: Log DB Error
E98: Function Not implement
E99: Unknown Error

(三) 檔案下載作業申請

BSTR NHI_Download(BSTR sSourcePath, BSTR
sBackfilePath, optional BSTR sDestName, optional BSTR
sTypeCode);

1. 函式功能說明：

- (1) 本函式將使用者端欲下載之檔案以參數檔型式傳送給健保資訊網，檔案格式依各作業類別定義，請參考該項作業說明。
- (2) 本函式每次傳送一個檔案，並將自動擷取醫療院所代碼及上傳作業日期時間。
- (3) 本函式作業若失敗，將回傳錯誤代碼。作業若成功，則傳回檔案執行編號，應用程式可依此代碼查詢作業情形。

2. 參數說明：(*) 為必填

- (1) * sSourcePath:原始檔案目錄及名稱
- (2) * sBackFileBame:下載檔案存放位置
- (3) sDestName:目的檔案名稱
- (4) *sTypeCode:定義的檔案種類

03：醫費申報資料檢核結果回饋資料

04：電子轉診資料下載_回復電子轉診單

- 05：預檢醫費申報資料檢核結果回饋資料
- 06：抽樣回饋資料
- 07：醫療費用電子申復資料檢核結果回饋資料
- 08：雲端藥歷下載
- 09：預檢醫療費用電子申復資料檢核結果回饋資料
- 26：檢驗(查)每日上傳資料 XML 格式檢核結果回饋資料
- 89：CT/MRI 檢核結果回饋資料

3. 回覆代碼：

- E01: Data/File Not Found Error
- E11: XML File Error
- E12: Copy File Error
- E13: Log DB Error
- E14:存放檔案的目錄不存在
- E98: Function Not implement
- E99: Unknown Error

(四) 檔案下載

BSTR NHI_Get(BSTR sOpcode);

1. 函式功能說明：

- (1) 本函式將先前透過 **NHI_Download** 函式進行下載申請的結果檔下載至 **NHI_Download** 函式指定的路徑。
- (2) 本函式作業若失敗，將回傳錯誤代碼。作業若成

功，則傳回 OK。

2. 參數說明：

- (1) nOpcode: NHI_Download 所回覆之有效執行編號，範例：NHI_45054353。

3. 回覆代碼：

E01: Data/File Not Found Error

A10: 上傳完成等待處理中

A11: 已受理等待處理中

A12: 檔案已就緒

F00: 重試次數過多

F10: 放棄執行

G03: 仍在處理中(MED)，請稍候

G04: 仍在處理中(EII)，請稍候

G07: 檔案接收失敗

G08: 檔案接收完成

E99: Unknown Error

(五) 查詢醫療資料檔案作業狀況

BSTR NHI_Query(BSTR sOpcode);

1. 函式功能說明：

- (1) 應用程式使用本函式查詢已交付傳送之檔案傳送作業狀況。
- (2) 使用本函式，需使用 NHI_Send、NHI_SendA 或 NHI_Download 所回復之執行編號作業。

2. 參數說明：

- (1) nOpcode: NHI_Send、NHI_SendA 或

NHI_Download 所回覆之有效執行編號，範例：

NHI_45054353。

3. VB.NET 範例：

```
Dim sRet as String
```

```
sRet = myObj.NHI_Query("NHI_45054353")
```

4. 回覆代碼：

A00: 已存檔、待傳送

A98: 傳送失敗

A10: 上傳完成等待處理中

A11: 已受理等待處理中

A12: 檔案已就緒

A13: 啟動下載作業

A14: 下載至院所端作業進行中

A99: 已傳送完成

B00: 認證中

B01: 認證完成，作業中

B02: 檔案加簽，作業中

B03: 作業傳送中

E01: 資料庫寫入異常

E02: 通訊異常

E04: 認證失敗

E05: 簽章失敗

E07: 認證失敗 1

E08: 認證失敗 2

E09: 認證失敗 3

E10: 認證失敗 4
E11: 認證失敗 5
E20: 傳送失敗 1
E21: 傳送失敗 2
E22: 傳送失敗 3
F00: 重試次數過多
F03: 檔案無法存取
F06: 檔案大小異常
F10: 放棄執行
F11: 檔案超出接收限定範圍
F12: 系統記憶體不足無法傳送該檔案
F20: 重試次數過多
G02: 連線異常
G03: 仍在處理中(MED)，請稍候
G04: 仍在處理中(EII)，請稍候
G05: 檔案建立失敗，請檢查
G06: 檔案寫入異常，請檢查
G07: 檔案接收失敗
G08: 檔案接收完成

(六) 刪除醫療資料檔案傳送作業

BSTR NHI_Delete(BSTR sOpcode);

1. 函式功能說明：

- (1) 本函式取消檔案之傳送作業。
- (2) 使用本函式，需使用 NHI_Send、NHI_SendA 或

NHI_Download 所回復之執行編號作業。

- (3) 若執行編號所代表之傳送作業尚未執行完成，本函式將立即取消該傳送作業，並回覆 OK。

2. 參數說明：

- (1) sOpcode: NHI_Send 、 NHI_SendA 或 NHI_Download 所回覆之有效執行編號，範例：
NHI_45054353。

3. 回覆代碼：

OK：刪除完成

FAIL：刪除失敗

玖、 API 使用範例

一、 Html 網頁

本範例為 COM 元件呼叫之範例，目前版本仍可繼續使用。

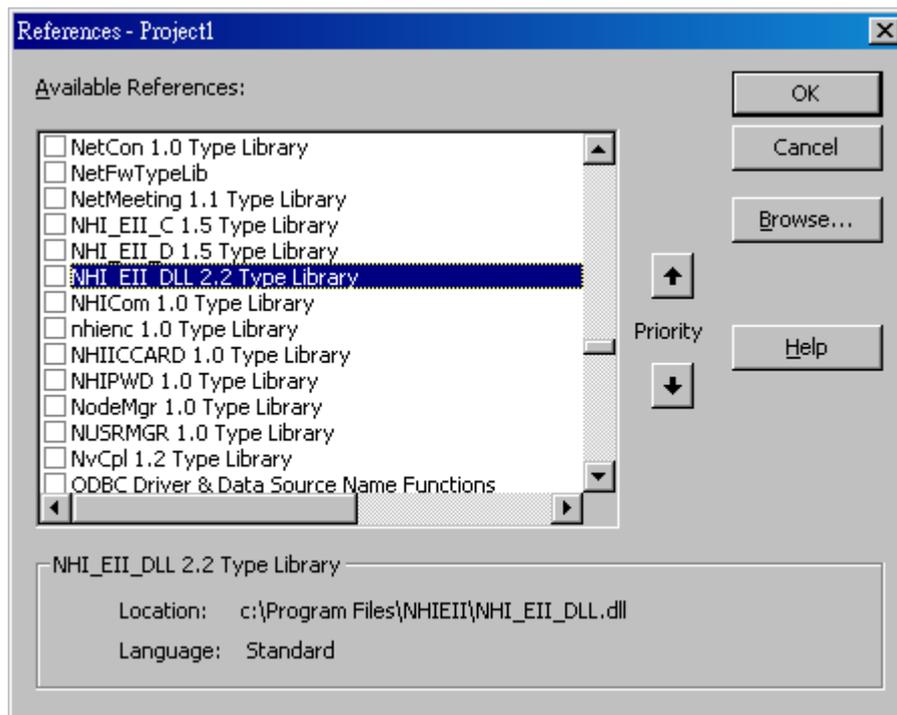
因應多元瀏覽器與 html5 之範例，請參考安裝目錄下的

C:\NHI\NHIEII\EIISample\EIISample.html

```
<HTML>
<HEAD>
<TITLE>ATL 3.0 test page for object NHI_EII_DLL</TITLE>
</HEAD>
<BODY>
<OBJECT ID="NHI_EII_DLL"
        CLASSID="CLSID:FB6C6FA1-C635-48AA-9056-0DB5C8D5F068"></OBJECT>
<SCRIPT language="JavaScript">

        function TestPACS_COM1()
        {
                if (NHI_EII_DLL!=null)
                {
                        alert(NHI_EII_DLL.NHI_Send("C:\\123.dat","1234567890"));
                        alert(NHI_EII_DLL.NHI_SendA("C:\\123.dat","1234567890","02");
                }
                else
                        alert("we not got com!");
        }
</SCRIPT>
<center> <input type="Button" value="測試 1" OnClick="TestPACS_COM1()">
</center>
</BODY>
</HTML>
```

二、VB 使用物件引用方式



‘宣告 NHI_EII_DLL 物件

```
Dim cNHI_PACS as NHI_EII_DLL
```

‘建立 NHI_EII_DLL 物件

```
Set cNHI_PACS = New NHI_EII_DLL
```

‘下例為立即傳送

```
sRet = cNHI_PACS.NHI_Send("C:\PACSDat1.dat",  
    "1Hosp12345620060816150000000")
```

```
sRet = cNHI_PACS.NHI_SendA("C:\PACSDat1.dat",  
    "1Hosp12345620060816150000000","02")
```

```
sRet = cNHI_PACS.NHI_Download("C:\Test\PACSDat1.dat",  
    "C:\Test",  
    "","05")
```

‘下例為晚間 11:00 傳送

```
sRet = cNHI_PACS.NHI_Send("C:\PACSDat2.dat",  
    "1Hosp12345620060816160000000","23:00:00")
```

‘結束 NHI_EII_EII 物件

```
Set cNHI_PACS = Nothin
```

三、 VB 使用 CreateObject 的呼叫方式

```
Dim MyObj As Object
Private Sub Command1_Click()

    Dim Ret As String
    Dim Ret1 As String
    Ret = MyObj.NHI_Send("C:\123.dat", "1234567890")
    Ret = MyObj.NHI_SendA("C:\123.dat", "1234567890", "02")
    Ret = MyObj.NHI_Download("C:\Test\123.dat", "c:\Test" & "", "05")
    MsgBox Ret
End Sub

Private Sub Form_Load()
    Set MyObj = CreateObject("NHI_EII_DLL.NHI_EII.1")
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set MyObj = Nothing
End Sub
```

四、VC++ 使用 CoCreateInstance 的呼叫方式

```
#include <comdef.h>
#include "NHI_EII_DLL.h" /* 附加於文件中 */
#include "NHI_EII_DLL_i.c" /* 附加於文件中 */
#include <atlbase.h>
void Test()
{
    HRESULT hr;
    hr=::CoInitialize(0);
    INHI_EII* pINHI_EII=NULL;
    hr = ::CoCreateInstance(CLSID_NHI_EII,
        NULL,
        CLSCTX_ALL,
        IID_INHI_EII,
        (void **)&pINHI_EII);
    if(SUCCEEDED(hr))
    {
        BSTR ret;
        hr = pINHI_EII->NHI_Send(L"c:\\123.dat",
            L"123456789012",
            L"2005/08/29 17:00:00",
            SEND_LOCAL,
            (unsigned short *)&ret);
        _bstr_t RetStr(ret);
        MessageBox(NULL, (char*)RetStr,"Info",MB_OK);
        hr = pINHI_EII->NHI_SendA(L"c:\\123.dat",
            L"123456789012",
            L"02",
            L"2005/08/29 17:00:00",
            SEND_LOCAL,
            (unsigned short *)&ret);
        _bstr_t RetStr(ret);
        MessageBox(NULL, (char*)RetStr,"Info",MB_OK);
    }
    pINHI_EII->Release();
    ::CoUninitialize();
}
```

五、VC++ 使用 Invoke 的呼叫方式

```
void Test()
{
    HRESULT hr;
    hr::CoInitialize(0);
    CLSID clsid;
    DISPID dispid;
    IDispatch* pIDispatch = NULL;
    wchar_t progid[]=L"NHI_EII_DLL.NHI_EII";
    OLECHAR* name=L"NHI_Send";
    CComVariant vRet;
    EXCEPINFO excepinfo;
    CComVariant vParam[2];
    vParam[0]=L"C:\\123.dat";
    vParam[1]=L"ABCDEFGH";
    DISPPARAMS DispParam = { vParam, NULL, 2, 0 };
    hr = ::CLSIDFromProgID(progid,&clsid);
    if(SUCCEEDED(hr))
    {
        hr = ::CoCreateInstance(clsid,NULL,CLSCTX_ALL,IID_IDispatch,(void**)&pIDispatch);
        if(SUCCEEDED(hr))
        {
            hr = pIDispatch->GetIDsOfNames(IID_NULL,
                                           &name,
                                           1,
                                           GetUserDefaultLCID(),
                                           &dispid);

            if(SUCCEEDED(hr))
            {
                hr = pIDispatch->Invoke(dispid,
                                       IID_NULL,
                                       GetUserDefaultLCID(),
                                       DISPATCH_METHOD,
                                       &DispParam,
                                       &vRet,
                                       NULL,
                                       NULL);

                _bstr_t RetStr(vRet);
                MessageBox(NULL, (char*)RetStr, "Info", MB_OK);
            }
        }
    }
}
```

```

        }
    }
}
::CoUninitialize();
}

```

```
void Test()
```

```

{
    HRESULT hr;
    hr=::CoInitialize(0);
    CLSID clsid;
    DISPID dispid;
    IDispatch* pIDispatch = NULL;
    wchar_t progid[]=L"NHI_EII_DLL.NHI_EII";
    OLECHAR* name=L"NHI_SendA";
    CComVariant vRet;
    EXCEPINFO excepinfo;
    CComVariant vParam[3];
    vParam[2]=L"C:\\123.dat";
    vParam[1]=L"ABCDEFGH";
    vParam[0]=L"02";
    DISPPARAMS DispParam = { vParam, NULL, 3, 0 };
    hr = ::CLSIDFromProgID(progid,&clsid);
    if(SUCCEEDED(hr))
    {
        hr = ::CoCreateInstance(clsid,NULL,CLSCTX_ALL,IID_IDispatch,(void**)&pIDispatch);
        if(SUCCEEDED(hr))
        {
            hr = pIDispatch->GetIDsOfNames(IID_NULL,
                                           &name,
                                           1,
                                           GetUserDefaultLCID(),
                                           &dispid);

            if(SUCCEEDED(hr))
            {
                hr = pIDispatch->Invoke(dispid,
                                       IID_NULL,
                                       GetUserDefaultLCID(),

```

```
DISPATCH_METHOD,  
&DispParam,  
&vRet,  
NULL,  
NULL);  
  
_bstr_t RetStr(vRet);  
MessageBox(NULL,(char*)RetStr,"Info",MB_OK);  
    }  
    }  
}  
::CoUninitialize();  
}
```

壹拾、附件

一、NHI_EII_DLL_i.c

```
/* ===== NHI_EII_DLL_i.c ===== */
/* NHI_EII_DLL_i.c 程式內容 */
/* this file contains the actual definitions of */
/* the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 5.01.0164 */
/* at Tue Aug 29 15:38:44 2006
*/
/* Compiler settings for C:\Develop\NHI_EII\src\NHI_EII_DLL\NHI_EII_DLL.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32, ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
*/
//@@@MIDL_FILE_HEADING( )
#ifdef __cplusplus
extern "C" {
#endif

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
};
#endif
```

```
} IID;
```

```
#endif // __IID_DEFINED__
```

```
#ifndef CLSID_DEFINED
```

```
#define CLSID_DEFINED
```

```
typedef IID CLSID;
```

```
#endif // CLSID_DEFINED
```

```
const IID IID_INHI_EII =
```

```
{0x64ECD190,0xDC77,0x4E7E,{0x88,0xE1,0x51,0x54,0x3F,0x59,0x76,0x72}};
```

```
const IID LIBID_NHI_EII_DLLLib =
```

```
{0xBB2805F5,0x4060,0x44FB,{0x95,0x89,0xD6,0xF5,0x20,0x10,0xA9,0x92}};
```

```
const CLSID CLSID_NHI_EII =
```

```
{0xFB6C6FA1,0xC635,0x48AA,{0x90,0x56,0x0D,0xB5,0xC8,0xD5,0xF0,0x68}};
```

```
#ifdef __cplusplus
```

```
}
```

```
#endif
```

二、 NHI_EII_DLL.h

```
/* ===== NHI_EII_DLL.h ===== */
/* NHI_EII_DLL.h 程式內容 */
/* this ALWAYS GENERATED file contains the definitions for the interfaces */
//@@MIDL_FILE_HEADING( )

/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __NHI_EII_DLL_h__
#define __NHI_EII_DLL_h__

#ifdef __cplusplus
extern "C"{
#endif

/* Forward Declarations */

#ifndef __INHI_EII_FWD_DEFINED__
#define __INHI_EII_FWD_DEFINED__
typedef interface INHI_EII INHI_EII;
#endif /* __INHI_EII_FWD_DEFINED__ */
```

```

#ifndef __NHI_EII_FWD_DEFINED__
#define __NHI_EII_FWD_DEFINED__

#ifdef __cplusplus
typedef class NHI_EII NHI_EII;
#else
typedef struct NHI_EII NHI_EII;
#endif /* __cplusplus */

#endif /* __NHI_EII_FWD_DEFINED__ */

/* header files for imported files */
#include "oidl.h"
#include "ocidl.h"

void __RPC_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* interface __MIDL_itf_NHI_EII_DLL_0000 */
/* [local] */

typedef /* [v1_enum] */
enum NHI_OpMethod
    {SEND_AUTH_SAM = 0x1,
      SEND_AUTH_HCA = 0x2,
      SEND_AUTH_HOC = 0x4,
      SEND_DESC_DEFAULT = 0x10,
      SEND_DESC_NONE = 0x20,
      SEND_COMP_NONE = 0x40,
      SEND_COMP_ZIP = 0x80,
      SEND_LOCAL = 0x100,
      SEND_REMOTE = 0x200
    }NHI_OpMethod;

extern RPC_IF_HANDLE __MIDL_itf_NHI_EII_DLL_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_NHI_EII_DLL_0000_v0_0_s_ifspec;

```

```

#ifndef __INHI_EII_INTERFACE_DEFINED__
#define __INHI_EII_INTERFACE_DEFINED__

/* interface INHI_EII */
/* [unique][helpstring][dual][uuid][object] */

EXTERN_C const IID IID_INHI_EII;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("64ECD190-DC77-4E7E-88E1-51543F597672")
    INHI_EII : public IDispatch
    {
    public:
        virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE NHI_Send(
            /* [in] */ BSTR sSourcePath,
            /* [in] */ BSTR sDestName,
            /* [defaultvalue][in][optional] */ BSTR sStartAt,
            /* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,
            /* [retval][out] */ BSTR __RPC_FAR *sOpCode) = 0;

        virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE HelloWorld(
            /* [retval][out] */ BSTR __RPC_FAR *ret) = 0;

        virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE GetVersion(
            /* [retval][out] */ BSTR __RPC_FAR *sVersion) = 0;

        virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE NHI_Query(
            /* [in] */ BSTR sOpCode,
            /* [retval][out] */ BSTR __RPC_FAR *sRetCode) = 0;

        virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE GetSetting(
            /* [in] */ BSTR sParameter,
            /* [retval][out] */ BSTR __RPC_FAR *sSetting) = 0;

        virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE NHI_Delete(
            /* [in] */ BSTR sOpCode,
            /* [retval][out] */ BSTR __RPC_FAR *sRetCode) = 0;

        virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE NHI_Change(

```

```

/* [in] */ BSTR sOpCode,
/* [defaultvalue][in][optional] */ BSTR tStartAt,
/* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,
/* [retval][out] */ BSTR __RPC_FAR *sRetCode) = 0;

```

```

virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE NHI_ResetJob(
/* [in] */ BSTR sOpCode,
/* [retval][out] */ BSTR __RPC_FAR *sRetCode) = 0;

```

```

virtual /* [helpstring][id] */ HRESULT STDMETHODCALLTYPE NHI_SendA(
/* [in] */ BSTR sSourcePath,
/* [in] */ BSTR sDestName,
/* [defaultvalue][in][optional] */ BSTR sTypeCode,
/* [defaultvalue][in][optional] */ BSTR sStartAt,
/* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,
/* [retval][out] */ BSTR __RPC_FAR *sOpCode) = 0;

```

```
};
```

```
#else /* C style interface */
```

```
typedef struct INHI_EIIVtbl
```

```
{
```

```
BEGIN_INTERFACE
```

```
HRESULT ( STDMETHODCALLTYPE __RPC_FAR *QueryInterface )(
```

```
INHI_EII __RPC_FAR * This,
```

```
/* [in] */ REFIID riid,
```

```
/* [iid_is][out] */ void __RPC_FAR * __RPC_FAR *ppvObject);
```

```
ULONG ( STDMETHODCALLTYPE __RPC_FAR *AddRef )(
```

```
INHI_EII __RPC_FAR * This);
```

```
ULONG ( STDMETHODCALLTYPE __RPC_FAR *Release )(
```

```
INHI_EII __RPC_FAR * This);
```

```
HRESULT ( STDMETHODCALLTYPE __RPC_FAR *GetTypeInfoCount )(
```

```
INHI_EII __RPC_FAR * This,
```

```
/* [out] */ UINT __RPC_FAR *pctinfo);
```

```
HRESULT ( STDMETHODCALLTYPE __RPC_FAR *GetTypeInfo )(
```

```

    INHI_EII __RPC_FAR * This,
    /* [in] */ UINT iTInfo,
    /* [in] */ LCID lcid,
    /* [out] */ ITypeInfo __RPC_FAR *__RPC_FAR *ppTInfo);

HRESULT ( STDMETHODCALLTYPE __RPC_FAR *GetIDsOfNames )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ REFIID riid,
    /* [size_is][in] */ LPOLESTR __RPC_FAR *rgszNames,
    /* [in] */ UINT cNames,
    /* [in] */ LCID lcid,
    /* [size_is][out] */ DISPID __RPC_FAR *rgDispId);

/* [local] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Invoke )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ DISPID dispIdMember,
    /* [in] */ REFIID riid,
    /* [in] */ LCID lcid,
    /* [in] */ WORD wFlags,
    /* [out][in] */ DISPPARAMS __RPC_FAR *pDispParams,
    /* [out] */ VARIANT __RPC_FAR *pVarResult,
    /* [out] */ EXCEPINFO __RPC_FAR *pExcepInfo,
    /* [out] */ UINT __RPC_FAR *puArgErr);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NHI_Send )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sSourcePath,
    /* [in] */ BSTR sDestName,
    /* [defaultvalue][in][optional] */ BSTR sStartAt,
    /* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,
    /* [retval][out] */ BSTR __RPC_FAR *sOpCode);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *HelloWorld )(
    INHI_EII __RPC_FAR * This,
    /* [retval][out] */ BSTR __RPC_FAR *ret);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *GetVersion )(
    INHI_EII __RPC_FAR * This,
    /* [retval][out] */ BSTR __RPC_FAR *sVersion);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NHI_Query )(

```

```

    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sOpCode,
    /* [retval][out] */ BSTR __RPC_FAR *sRetCode);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *GetSetting )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sParameter,
    /* [retval][out] */ BSTR __RPC_FAR *sSetting);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NHI_Delete )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sOpCode,
    /* [retval][out] */ BSTR __RPC_FAR *sRetCode);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NHI_Change )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sOpCode,
    /* [defaultvalue][in][optional] */ BSTR tStartAt,
    /* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,
    /* [retval][out] */ BSTR __RPC_FAR *sRetCode);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NHI_ResetJob )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sOpCode,
    /* [retval][out] */ BSTR __RPC_FAR *sRetCode);

/* [helpstring][id] */ HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NHI_SendA )(
    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sSourcePath,
    /* [in] */ BSTR sDestName,
    /* [defaultvalue][in][optional] */ BSTR sTypeCode,
    /* [defaultvalue][in][optional] */ BSTR sStartAt,
    /* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,
    /* [retval][out] */ BSTR __RPC_FAR *sOpCode);

    END_INTERFACE
} INHI_EIIVtbl;

interface INHI_EII
{
    CONST_VTBL struct INHI_EIIVtbl __RPC_FAR *lpVtbl;

```

```
};
```

```
#ifdef COBJMACROS
```

```
#define INHI_EII_QueryInterface(This,riid,ppvObject) \  
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)
```

```
#define INHI_EII_AddRef(This) \  
    (This)->lpVtbl -> AddRef(This)
```

```
#define INHI_EII_Release(This) \  
    (This)->lpVtbl -> Release(This)
```

```
#define INHI_EII_GetTypeInfoCount(This,pctinfo) \  
    (This)->lpVtbl -> GetTypeInfoCount(This,pctinfo)
```

```
#define INHI_EII_GetTypeInfo(This,iTInfo,lcid,ppTInfo) \  
    (This)->lpVtbl -> GetTypeInfo(This,iTInfo,lcid,ppTInfo)
```

```
#define INHI_EII_GetIDsOfNames(This,riid,rgszNames,cNames,lcid,rgDispld) \  
    (This)->lpVtbl -> GetIDsOfNames(This,riid,rgszNames,cNames,lcid,rgDispld)
```

```
#define INHI_EII_Invoke(This,displdMember,riid,lcid,wFlags,pDispParams,pVarResult,pExceplInfo,puArgErr) \  
    \  
    (This)->lpVtbl -> Invoke(This,displdMember,riid,lcid,wFlags,pDispParams,pVarResult,pExceplInfo,puArgErr)
```

```
#define INHI_EII_NHI_Send(This,sSourcePath,sDestName,sStartAt,nOpMethod,sOpCode) \  
    (This)->lpVtbl -> NHI_Send(This,sSourcePath,sDestName,sStartAt,nOpMethod,sOpCode)
```

```
#define INHI_EII_HelloWorld(This,ret) \  
    (This)->lpVtbl -> HelloWorld(This,ret)
```

```
#define INHI_EII_GetVersion(This,sVersion) \  
    (This)->lpVtbl -> GetVersion(This,sVersion)
```

```

#define INHI_EII_NHI_Query(This,sOpCode,sRetCode) \
    (This)->lpVtbl -> NHI_Query(This,sOpCode,sRetCode)

#define INHI_EII_GetSetting(This,sParameter,sSetting) \
    (This)->lpVtbl -> GetSetting(This,sParameter,sSetting)

#define INHI_EII_NHI_Delete(This,sOpCode,sRetCode) \
    (This)->lpVtbl -> NHI_Delete(This,sOpCode,sRetCode)

#define INHI_EII_NHI_Change(This,sOpCode,tStartAt,nOpMethod,sRetCode) \
    (This)->lpVtbl -> NHI_Change(This,sOpCode,tStartAt,nOpMethod,sRetCode)

#define INHI_EII_NHI_ResetJob(This,sOpCode,sRetCode) \
    (This)->lpVtbl -> NHI_ResetJob(This,sOpCode,sRetCode)

#define INHI_EII_NHI_SendA(This,sSourcePath,sDestName,sTypeCode,sStartAt,nOpMethod,sOpCode) \
    \
    (This)->lpVtbl -> NHI_SendA(This,sSourcePath,sDestName,sTypeCode,sStartAt,nOpMethod,sOpCode)

#endif /* COBJMACROS */

#endif /* C style interface */

/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_NHI_Send_Proxy(
    INHI_EII __RPC_FAR * This,
    /* [in] */ BSTR sSourcePath,
    /* [in] */ BSTR sDestName,
    /* [defaultvalue][in][optional] */ BSTR sStartAt,
    /* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,
    /* [retval][out] */ BSTR __RPC_FAR *sOpCode);

void __RPC_STUB INHI_EII_NHI_Send_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_HelloWorld_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [retval][out] */ BSTR __RPC_FAR *ret);
```

```
void __RPC_STUB INHI_EII_HelloWorld_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_GetVersion_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [retval][out] */ BSTR __RPC_FAR *sVersion);
```

```
void __RPC_STUB INHI_EII_GetVersion_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_NHI_Query_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [in] */ BSTR sOpCode,  
    /* [retval][out] */ BSTR __RPC_FAR *sRetCode);
```

```
void __RPC_STUB INHI_EII_NHI_Query_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_GetSetting_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [in] */ BSTR sParameter,
```

```
/* [retval][out] */ BSTR __RPC_FAR *sSetting);
```

```
void __RPC_STUB INHI_EII_GetSetting_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_NHI_Delete_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [in] */ BSTR sOpCode,  
    /* [retval][out] */ BSTR __RPC_FAR *sRetCode);
```

```
void __RPC_STUB INHI_EII_NHI_Delete_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_NHI_Change_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [in] */ BSTR sOpCode,  
    /* [defaultvalue][in][optional] */ BSTR tStartAt,  
    /* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,  
    /* [retval][out] */ BSTR __RPC_FAR *sRetCode);
```

```
void __RPC_STUB INHI_EII_NHI_Change_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_NHI_ResetJob_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [in] */ BSTR sOpCode,
```

```
/* [retval][out] */ BSTR __RPC_FAR *sRetCode);
```

```
void __RPC_STUB INHI_EII_NHI_ResetJob_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
/* [helpstring][id] */ HRESULT STDMETHODCALLTYPE INHI_EII_NHI_SendA_Proxy(  
    INHI_EII __RPC_FAR * This,  
    /* [in] */ BSTR sSourcePath,  
    /* [in] */ BSTR sDestName,  
    /* [defaultvalue][in][optional] */ BSTR sTypeCode,  
    /* [defaultvalue][in][optional] */ BSTR sStartAt,  
    /* [defaultvalue][in][optional] */ NHI_OpMethod nOpMethod,  
    /* [retval][out] */ BSTR __RPC_FAR *sOpCode);
```

```
void __RPC_STUB INHI_EII_NHI_SendA_Stub(  
    IRpcStubBuffer *This,  
    IRpcChannelBuffer *_pRpcChannelBuffer,  
    PRPC_MESSAGE _pRpcMessage,  
    DWORD *_pdwStubPhase);
```

```
#endif /* __INHI_EII_INTERFACE_DEFINED__ */
```

```
#ifndef __NHI_EII_DLLlib_LIBRARY_DEFINED__  
#define __NHI_EII_DLLlib_LIBRARY_DEFINED__
```

```
/* library NHI_EII_DLLlib */  
/* [helpstring][version][uuid] */
```

```
EXTERN_C const IID LIBID_NHI_EII_DLLlib;
```

```
EXTERN_C const CLSID CLSID_NHI_EII;
```

```
#ifdef __cplusplus
```

```
class DECLSPEC_UUID("FB6C6FA1-C635-48AA-9056-0DB5C8D5F068")
```

```
NHI_EII;
```

```
#endif
```

```
#endif /* __NHI_EII_DLLLib_LIBRARY_DEFINED__ */
```

```
/* Additional Prototypes for ALL interfaces */
```

```
unsigned long          __RPC_USER  BSTR_UserSize(      unsigned long __RPC_FAR *, unsigned  
    long                , BSTR __RPC_FAR * );
```

```
unsigned char __RPC_FAR * __RPC_USER  BSTR_UserMarshal(  unsigned long __RPC_FAR *,  
    unsigned char __RPC_FAR *, BSTR __RPC_FAR * );
```

```
unsigned char __RPC_FAR * __RPC_USER  BSTR_UserUnmarshal(unsigned long __RPC_FAR *,  
    unsigned char __RPC_FAR *, BSTR __RPC_FAR * );
```

```
void            __RPC_USER  BSTR_UserFree(      unsigned long __RPC_FAR *, BSTR  
    __RPC_FAR * );
```

```
/* end of Additional Prototypes */
```

```
#ifdef __cplusplus
```

```
}
```

```
#endif
```

```
#endif
```