Single Sign On 順向代簽入程式導入相關資訊

說明「順向代簽入」:

導入程式 readssoCcuRightXML.php + refreshSso.php

常數：

SYS_DOOR_URL	子系統端登入頁
SYS_LOGIN_URL	子系統端登入驗證判斷程式
SSO_DOOR_URL	SSO 端首頁

重要 SESSION 變數：

$_SESSION['verifySso']	SSO 登入方式辨識訊號
$_SESSION['tokenSso']	SSO 登入 token 資訊
$_SESSION['sso_enterip']	使用者端登入 IP
$_SESSION['sso_personid']	身份證字號

函式：

sso_getIP()	取得使用者 IP
chk_ssoRight($mix_info)	取得 sso 權限資訊
chk_ssoRefresh($mix_info)	延長 sso 的 session 存活時間
err_msgAlert($msg,$url)	錯誤訊息及導向頁面
ssoLogOut()	SSO 的登出導向

導入步驟

1　設定 URL 常數

➔ refreshSso.php 的 line33,line34

```
define('SYS_DOOR_URL',"子系統端登入頁");
define('SYS_LOGIN_URL',"子系統端登入驗證判斷程式");
```

2　在登入驗證判斷程式(例如 login.php)中 include 導入程式 readssoCcuRightXML.php

```
include 'readssoCcuRightXML.php';
```

3 導入初期，使用雙軌並行！因此在登入驗證判斷中，須判別是從原來的登入方式登入，還是從 SSO 的登入方式登入。

利用 $_SESSION['verifySso'] 做為 SSO 端登入方式的辨識訊號。

因此在登入驗證判斷中除了接收身份證字號外，必需區隔判別其登入方式。

若是從 SSO 接收到登入資訊者，$_SESSION['verifySso'] 變數會存在，且被設為字串'Y'。

以某系統為例，在登入驗證判斷程式中判斷如下：

```
if( empty($_POST)   AND   isset($_SESSION['verifySso'])   AND   $_SESSION['verifySso']=='Y' )
{
      //sso 代簽入的 code
      //接收 $_SESSION['sso_personid'] (身份證字號)，做登入轉換
}
else
{
      //原本的登入機制
}
```

4 由登入驗證判斷中，若 isset($_SESSION['verifySso'])成立，則接收 $_SESSION['sso_personid'] (身份證字號)，再做登入轉換。

這部分各系統狀況不同，由各系統想辦法轉換。

5 在每個程式頁面 include 導入程式 refreshSso.php

目的是在判別 isset($_SESSION['verifySso']) 成立後，執行 chk_ssoRefresh($_SESSION['tokenSso'])函式，以便延長 SSO 端的 session 存活時間。(詳參閱 refreshSso.php, line 125)

6 登出程式多加一個判斷。

若是從原有登入方式登入者，登出作法維持原狀。

若是從 SSO 端登入方式登入者，登出作法改變為 SSO 的登出方式。

```
if( isset($_SESSION[‘verifySso’]) ){
      ssoLogOut();      //SSO 的登出方式
}else{
      ……              //原有的登出方式
}
```

7 修改後測試下列情況，如果都正常就 ok 了

7.1 直接從子系統登入登出(也就是最原本的方式)

7.2 直接從 sso 代簽入，代簽入成功應該在子系統登入成功後的頁面，登出後則是回到 sso 的頁面

7.3 子系統先用原本方法登入，再從 sso 端登入，代簽入成功應該在子系統登入成功後的頁面，登出後則是回到 sso 的頁面(跟第 2 點一樣)

7.4 sso 代簽入到子系統後，另外開新分頁，直接輸入子系統網址，此時應該要直接帶到登入成功的頁面(而不是跳出原本子系統的登入頁面)

說明「逆向代簽入」➔開發中尚未完成