A brief documentation of SessionChange Service

❖ **Introduction:** This is a Windows service example based on arcker's UDF. This service captures the *SERVICE_CONTROL_SESSIONCHANGE* event as described in HandlerEx documentation. This event triggers a WM_WTSSESSION_CHANGE message whose handler function accepts four parameters. The handler function is none but the HandlerEx callback function already registered by a call to RegisterServiceCtrlHandlerEx. The third parameter of HandlerEx function is of our interest as; the system sends one or more of the nine messages depending upon the session change event that has taken place, viz. *WTS_CONSOLE_CONNECT, WTS_CONSOLE_DISCONNECT, WTS_REMOTE_CONNECT, WTS_REMOTE_DISCONNECT, WTS_SESSION_LOGON, WTS_SESSION_LOGOFF, WTS_SESSION_LOCK, WTS_SESSION_UNLOCK & WTS_SESSION_REMOTE_CONTROL.* The second parameter of HandlerEx is of little less importance. It contains a pointer to WTSSESSION_NOTIFICATION structure which, in turn, embeds the session ID of the user who has caused the session change event to take place.

❖ **Interesting code snippets:** Here are a few code snippets which I've added to **arcker**'s UDF.

1. File: ServicesConstants.au3

   Code:

```
; Service event types
Global Const $WTS_CONSOLE_CONNECT = 0x00000001
Global Const $WTS_CONSOLE_DISCONNECT = 0x00000002
Global Const $WTS_REMOTE_CONNECT = 0x00000003
Global Const $WTS_REMOTE_DISCONNECT = 0x00000004
Global Const $WTS_SESSION_LOGON = 0x00000005
Global Const $WTS_SESSION_LOGOFF = 0x00000006
Global Const $WTS_SESSION_LOCK = 0x00000007
Global Const $WTS_SESSION_UNLOCK = 0x00000008
Global Const $WTS_SESSION_REMOTE_CONTROL = 0x00000009
```

   Explanation: Self explanatory

2. File: Services.au3

   Function: _Service_Ctrl()

Code:

```
Case $SERVICE_CONTROL_SESSIONCHANGE
; Extracting SessionID & cbSize
Local $WTSSESSION_NOTIFICATION  = DllStructCreate("dword cbsize; dword dwSessionId", $lpEventData)
Local $cbSize = DllStructGetData($WTSSESSION_NOTIFICATION, "cbSize")
Local $dwSessionId = DllStructGetData($WTSSESSION_NOTIFICATION, "dwSessionId")
WriteLog("cbSize = " & $cbSize & " ,dwSessionId = " & $dwSessionId)

;Trapping events
Switch $dwEventType
    Case $WTS_CONSOLE_CONNECT
        WriteLog("Console session connected", 0, 2)
    Case $WTS_CONSOLE_DISCONNECT
        WriteLog("Console session disconnected", 0, 2)
    Case $WTS_REMOTE_CONNECT
        WriteLog("Remote session connected", 0, 2)
    Case $WTS_REMOTE_DISCONNECT
        WriteLog("Remote session disconnected", 0, 2)
    Case $WTS_SESSION_LOGON
        WriteLog("Session logged on", 0, 2)
    Case $WTS_SESSION_LOGOFF
        WriteLog("Session logged off", 0, 2)
    Case $WTS_SESSION_LOCK
        WriteLog("Session locked", 0, 2)
    Case $WTS_SESSION_UNLOCK
        WriteLog("Session unlocked", 0, 2)
    Case $WTS_SESSION_REMOTE_CONTROL
        WriteLog("Session remote control", 0, 2)
    EndSwitch
```

Explanation: _Service_Ctrl() function in **arcker**'s UDF serves the purpose of **HandlerEx** callback function as stated in MSDN. It was previously registered to handle all service messages during the creation of the service. **$lpEventData** contains a pointer to WTSSESSION_NOTIFICATION structure. **DllStructCreate()** creates a structure of the mentioned type & passed the pointer **$lpEventData** to populate the newly created structure with the data received from the system. Subsequent calls to **DllStructGetData()** retrieves **$cbSize** & **$dwSessionId**. **$dwEventType** is tested against the types of possible session change events.

3. File: Services.au3

   Function: _Service_ReportStatus()

   Code:

   ```
   DllStructSetData($tService_Status, "dwControlsAccepted", BitOR($SERVICE_ACCEPT_STOP, $SERVICE_ACCEPT_SESSIONCHANGE))
   ```

   Explanation: **BitOR()-**ing announces that the service can accept both **SERVICE_ACCEPT_STOP** & **SERVICE_ACCEPT_SESSIONCHANGE** messages.

❖ Contents of the folder:

/Scripts/SessionChange.au3 : - The entry point of the service script
/Scripts/Services.au3: - Reduced version of **arcker**'s UDF
/Scripts/ServicesConstants.au3: - All the constants required
/Scripts/Log.au3: - Contains log generating functions

/Logs/SessionChange_27.05.2011 _ 19.18.50.log : - A sample log while installing service
/Logs/SessionChange_27.05.2011 _ 19.18.53.log : - A sample log while running service
/Logs/SessionChange_27.05.2011 _ 19.18.41.log: - A sample log while removing service

❖ Working with the service:

1. Compile **SessionChange.au3** in non-Console/GUI mode.
2. Double click on **SessionChange.au3.exe**/ pass **SessionChange –i** from command line to install the service.
3. Click "Yes" on "Start running service?" dialog / Click Control Panel -> Administrative Tools -> Services -> SessionChange -> Start/ Pass **SessionChange –r** from command line to start the service.
4. Click Control Panel -> Administrative Tools -> Services -> SessionChange -> Stop/ Pass **SessionChange –s** from command line to stop the service.
5. Pass **SessionChange –u** from command line to uninstall the service.
6. **DO NOT** attempt to test the code directly from SciTe.

Regards: - HolmesSherlock