

```
#include <IE.au3>
#Autolt3Wrapper_UseX64=n
Opt("MustDeclareVars", 1)
AutoltSetOption("WinTitleMatchMode", 2) ; EXACT_MATCH!
```

```
=====
;
;   PSM Autolt Dispatcher Skeleton
;   -----
;
; Use this skeleton to create your own
; connection components integrated with the PSM.
; Areas you may want to modify are marked
; with the string "CHANGE_ME".
;
; Created : April 2013
; Cyber-Ark Software Ltd.
=====
```

```
#include "PSMGenericClientWrapper.au3"
```

```
=====
; Consts & Globals
=====
```

```
Global Const $DISPATCHER_NAME           = "PSMABCDispatcher" ; CHANGE_ME
Global Const $LOGIN_WEBPAGE              = "web url" ; CHANGE_ME
Global Const $CLIENT_EXECUTABLE         = "c:\Program Files (x86)\Internet Explorer\iexplore.exe" &
Global Const $ERROR_MESSAGE_TITLE       = "PSM " & $DISPATCHER_NAME & " Dispatcher error"
Global Const $LOG_MESSAGE_PREFIX        = $DISPATCHER_NAME & " Dispatcher - "
```

```
Global $TargetUsername
Global $TargetPassword
Global $TargetAddress
Global $ConnectionClientPID = 0
Global $oIE
```

```
=====
; Code
=====
Exit Main()
```

```
=====
; Main
=====
Func Main()
```

```
    ; Init PSM Dispatcher utils wrapper
    ToolTip ("Initializing...")
    if (PSMGenericClient_Init() <> $PSM_ERROR_SUCCESS) Then
        Error(PSMGenericClient_PSMGetLastErrorString())
    EndIf
```

```
    LogWrite("successfully initialized Dispatcher Utils Wrapper")
```

```
    ; Get the dispatcher parameters
    FetchSessionProperties()
```

```
    LogWrite("starting client application")
    ToolTip ("Starting " & $DISPATCHER_NAME & "...")
```

```

$oIE = _IECreate($LOGIN_WEBPAGE, 0, 0) ; Remove the zeros if not hardening
HardenIE()
_IEAction($oIE, "visible")

; Get the IE Handle ID
dim $hndl = _IEPropertyGet($oIE, "hwnd")
$ConnectionClientPID = WinGetProcess($hndl)
;$ConnectionClientPID = Run($CLIENT_EXECUTABLE)
if ($ConnectionClientPID == 0) Then
    Error(StringFormat("Failed to execute process [%s]", $CLIENT_EXECUTABLE, @error))
EndIf

; Send PID to PSM as early as possible so recording/monitoring can begin
LogWrite("sending PID to PSM")
if (PSMGenericClient_SendPID($ConnectionClientPID) <> $PSM_ERROR_SUCCESS) Then
    Error(PSMGenericClient_PSMGetLastErrorString())
EndIf

; IELinkClickByText($oIE, "Continue to this website (not recommended).")

; -----
; Handle login here! ; CHANGE_ME
; -----
Local $o_form = _IEFormGetObjByName ($oIE, "logonForm")
;Local $o_Form = _IEFormGetCollection ($oIE, 0)
Local $o_user = _IEFormElementGetObjByName ($o_form, "j_username")
Local $o_password = _IEFormElementGetObjByName ($o_form, "j_password")
Local $o_signin = _IEFormElementGetObjByName ($o_form, "Login")

; Set field values and submit the form
_IEFormElementSetValue ($o_user, $TargetUsername)
_IEFormElementSetValue ($o_password, $TargetPassword)
_IEAction ($o_signin, "click")

; Terminate PSM Dispatcher utils wrapper
LogWrite("Terminating Dispatcher Utils Wrapper")
PSMGenericClient_Term()

Return $PSM_ERROR_SUCCESS
EndFunc

;=====
; Functions
;=====
; #FUNCTION# -----
; Name.....: Error
; Description ....: An exception handler - displays an error message and terminates the dispatcher
; Parameters ....: $ErrorMessage - Error message to display
;                 $Code       - [Optional] Exit error code
;=====
Func Error($ErrorMessage, $Code = -1)

; If the dispatcher utils DLL was already initialized, write an error log message and terminate the wrapper
if (PSMGenericClient_IsInitialized()) Then
    LogWrite($ErrorMessage, True)
    PSMGenericClient_Term()
EndIf

```

```

Local $MessageFlags = BitOr(0, 16, 262144) ; 0=OK button, 16=Stop-sign icon, 262144=MsgBox has top-most attri

MsgBox($MessageFlags, $ERROR_MESSAGE_TITLE, $ErrorMessage)

; If the connection component was already invoked, terminate it
if ($ConnectionClientPID <> 0) Then
    ProcessClose($ConnectionClientPID)
    $ConnectionClientPID = 0
EndIf

Exit $Code
EndFunc

;#FUNCTION# =====
; Name.....: LogWrite
; Description ...: Write a PSMWinSCPDispatcher log message to standard PSM log file
; Parameters ....: $sMessage - [IN] The message to write
;                 $LogLevel - [Optional] [IN] Defined if the message should be handled as an error message or as a trace m
; Return values ..: $PSM_ERROR_SUCCESS - Success, otherwise error - Use PSMGenericClient_PSMGetLastErrorS
; =====
Func LogWrite($sMessage, $LogLevel = $LOG_LEVEL_TRACE)
    Return PSMGenericClient_LogWrite($LOG_MESSAGE_PREFIX & $sMessage, $LogLevel)
EndFunc

;#FUNCTION# =====
; Name.....: PSMGenericClient_GetSessionProperty
; Description ...: Fetches properties required for the session
; Parameters ....: None
; Return values ..: None
; =====
Func FetchSessionProperties() ; CHANGE_ME
    if (PSMGenericClient_GetSessionProperty("Username", $TargetUsername) <> $PSM_ERROR_SUCCESS) Then
        Error(PSMGenericClient_PSMGetLastErrorString())
    EndIf

    if (PSMGenericClient_GetSessionProperty("Password", $TargetPassword) <> $PSM_ERROR_SUCCESS) Then
        Error(PSMGenericClient_PSMGetLastErrorString())
    EndIf

    if (PSMGenericClient_GetSessionProperty("Address", $TargetAddress) <> $PSM_ERROR_SUCCESS) Then
        Error(PSMGenericClient_PSMGetLastErrorString())
    EndIf
EndFunc

Func SetIEProperty($sProperty, $nValue)
    dim $oIE
    if(_IEPropertySet($oIE, $sProperty, $nValue) <> 0) Then
        Error(StringFormat("WebFormDispatcher: Failed to set IE property [%s] to [%s] (Extra details: %d, %d)", $sPrope
    EndIf
EndFunc

Func HardenIE()
    SetIEProperty("fullscreen", 1)
    SetIEProperty("theatermode", 1)
    SetIEProperty("toolbar", 0)
    SetIEProperty("menubar", 0)
    SetIEProperty("Addressbar", 0)
EndFunc

```