

Session_TO Manual (version 1.4 - 20130530)

1 - Introduction

This is an OCX control that enables you to compute the time elapsed in frame based web application / pages, so you can take a redirection action or close the browser at the time out condition without interaction with the server.

These allows to include this session timeout control in application where there is no application servers involved or a timer schema, by providing a low effort re-engineering to add a time control to an existing application.

Once installed and initialized there is a timer that can be reset with each mouse or keyboard event; but only when the focus is on the specific browser instance.

This enabled e.g. the possibility of having an automatic finish of an application when there is no activity in the associated browser windows - where the object was installed or child windows generated from this.

This control also permit to close the browser window associated with the navigation of a specific site/application when the time expires.

2 – Getting Requirements

2.1

This control **requires almost IExplore version 6**. At the time of this publication (control version 1.4) it has been tested with IExplore 7, 8 and 9 versions.

It is required to configure IExplore ActiveX security parameters in Internet option to allow the component operation.

Please refer to section 4 to observe the demo limitations

2.2 Installation Instructions

- a) Download the package. from www.sessioto.com



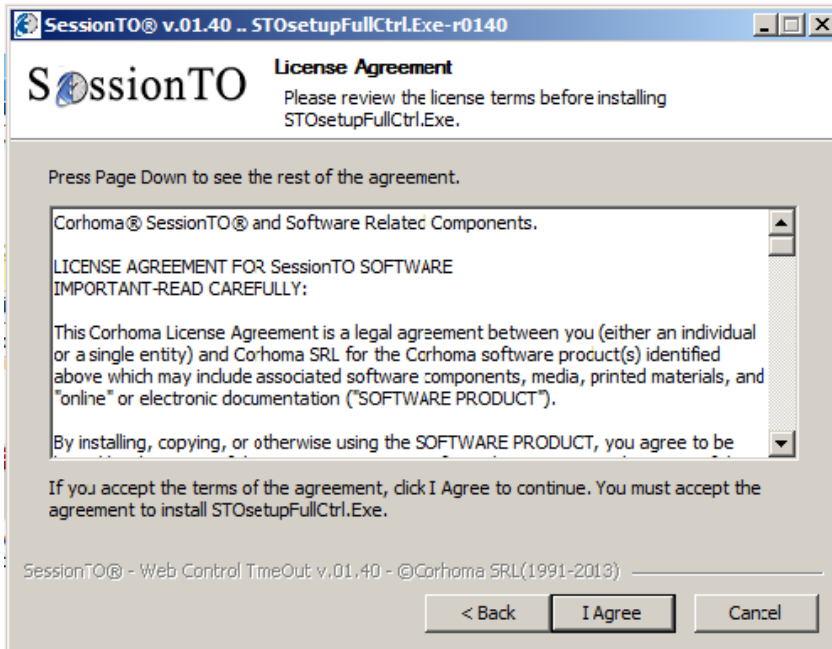
- b) Install the distribution

Once you have received the key, proceed to do the decrypt the software package. You will get following executable: STOsetupFullCtrl.exe

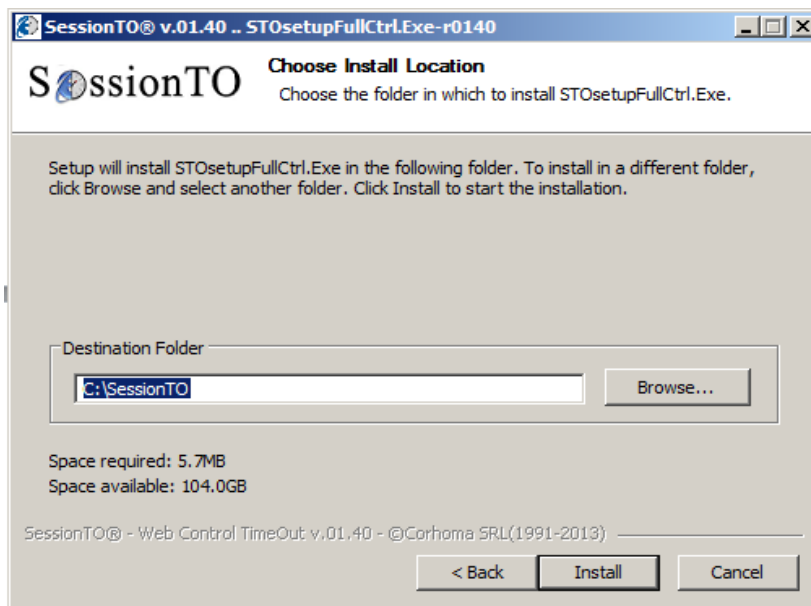
It will lead you to copy all the modules and components for the distribution. When you execute the program It shows following sequence:



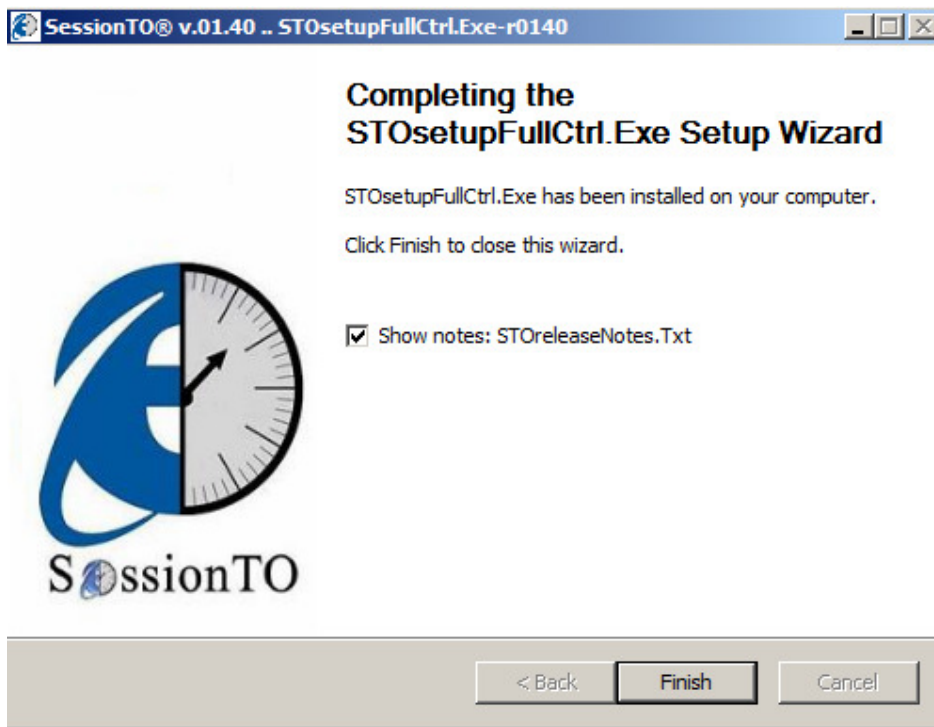
By pressing next you will read the license agreement :



If you agree with the license terms, it will prompt you to enter the path



And after copying object, manual, sample and notes it will finish



Now you can get the .CAB module from the bin folder .

That component should be copied where the web application will upload the component from.

It is required less than 15 MB storage space where the software will be expanded.

3 - Code Examples

There are sample control implementations – source code included - in both demo and full packages.

There is an online sample included in the **SessionTO** web site samples demo link.

All code examples are coded in javascript.

4 - Differences between the demo and production version

Following are **the limitations for the demo** version:

- Session time fixed to 60 (sixty) seconds
- Control is no digitally signed
- Fixed redirection to exit test page

5 - Security Settings in Internet Explorer

There is no special consideration to be done.

You have to install the Corhoma root public certificate and the object to get the desired object capabilities.

As other **Activex** controls you can set the proper security restriction by enabling / disable in in browser tools / internet options / Security tag, the topics concerning activex download, install and execution.

In such case consult your security administrator

Remember the DEMO object IS NOT A SIGNED component .

6 - Operation

There is a programmed hook for mouse and keyboard event , so once the session timeout object was initialized. an elapsed time computing process takes place and might be reset by the keyboard or mouse events in that browser instance scope (if enabled).

Once the time period is reached it is possible to display a message announcing the T.O. condition and finish with two types of ending sequences: closing the browser or redirecting the current page to other specified previously by program object capabilities..

All object activities can be logged to a file to allow a subsequent event analysis.

7 - Getting started

Note:

All the examples will be referenced to a control with an known ID - we will use "**TimeOutCtl**" as the object ID in this examples, and the language used to implement the sample code will be javascript.

To explain how to use this control we will assume a pre-existing frames based web application where we want control the time expiration though by user inactivity in that scope or just for a pre-set time (as programmed)

Example 1

Problem: **Closing the browser after 3 minutes of starting the page**

a) Download and copy the control

Copy the **session_TO.cab** distributed file in the directory where the page expect find the object.

b) Declaring the object

In the initial page were you want to want to star computing time include following reference:

```
<OBJECT ID="TimeOutCtl"  
  
CLASSID="CLSID:EA1C0DD7-7988-4E00-A81E-2858F30BD627"  
  
CODEBASE="crh_web_ses_TO.CAB#version=1,4,0,0">  
  
</OBJECT>
```

In this reference CLASSID is an example . Please use the declaration included in the sample code (you must do that to have the proper one at the integration time)

After including the object you have to do the object setting and then start the time accounting.

c) Setup and initialize object

You can include such a piece of javascript code:

```
<script LANGUAGE="JavaScript">

// -----
// sample 1
//-----

var vcrh_ws_TO_msg = " Time OUT - sample end"; // text to be displayed
                                                // in the message box

var vcrh_ws_TO_time = 120 ;                    // session period in seconds
var vcrh_ws_TO_qsc = 2 ;                      // quantity exist comands
var vcrh_ws_TO_show_msgTO = 1;                // display T.O. msgbox when
                                                // T.O. occurs

var vcrh_ws_TO_nores = 1                      // don't clear elapsed time
var vcrh_app_exitmet = 1                      // finish ending browser
                                                // ( close it )

document.TimeOutCtl.Set_STO_To_msg(vcrh_ws_TO_msg); // set the T.O.
                                                    // msg instead
                                                    // default msg

document.TimeOutCtl.Set_STO_Time( vcrh_ws_TO_time ); // set the time

document.TimeOutCtl.Set_STO_Exit_Method(vcrh_app_exitmet); // set the
                                                            //exit
                                                            /method to
                                                            end

document.TimeOutCtl.Set_STO_qsc( vcrh_ws_TO_qsc ); // set quantity
of exit commnads

document.TimeOutCtl.Show_STO_Msg(vcrh_ws_TO_show_msgTO) // enable
                                                            // display
                                                            // T.O. msg
```

```

document.Set_STO_NoResTime( vcrh_ws_TO_nores) //clear elapsed time
                                                //if keyb / mouse
                                                //events

document.TimeOutCtl.Start_STO() // Start time computing ...

</script>

```

The object will compute time elapsed. No matter if there mouse clicks events or keyboard pressed events in the instance browser scope, there will appear a message box after two minutes (120 SECONDS) with the programmed legend " Time OUT - sample end". When you click on the message box button the browser will be closed.

See SAMPLE1 folder for the source code sample

Example 2

Problem: redirect the application after 30 minutes of inactivity so we can get information which station was and who where logged finishing the by inactivity

a) Download and copy

Copy the **session_TO.cab** distributed file in the directory where the page expect find the object.

b) Declaring the object

in the initial page were you want to want to star computing time include following reference :

```

<OBJECT ID="TimeOutCtl"

        CLASSID="CLSID:EA1C0DD7-7988-4E00-A81E-2858F30BD627"

        CODEBASE="session_TO.CAB#version=1,4,0,0">

</OBJECT>

```


In this reference CLASSID is an example . Use the declaration included in the sample code to have the proper one at the integration time.

After including the object you have to do the object setting, and then start the time accounting.

c) Setup and initialize object

You can include this a piece of javascript code like this:

```
<script LANGUAGE="JavaScript">

// -----
//      sample 2
// -----

// wher to go ...
var vcrh_ws_TO_url2go = "http://serverbckp:8090/crhadmin/exit.php";

var vcrh_ws_TO_time = 1800 ;           // time in seconds
var vcrh_ws_TO_nores = 0; // 0 = clear elapsed time each keyb / mouse
// event
var vcrh_ws_TO_show_msgTO = 0;       // no display msg T.O.
var vcrh_app_exitmet = 3 ;           // redir std
var sz_str_redir_prebuf = "{F11}%D{DEL}" ;// cmd POINT MOUSE TO ADD
BAR

document.TimeOutCtl.Set_STO_Url_2_Go(vcrh_ws_TO_url2go); // set URL
// to go if
// Time Out

document.TimeOutCtl.Set_STO_Time( vcrh_ws_TO_time ); //set T.O. period

document.TimeOutCtl.Set_STO_Exit_Method(vcrh_app_exitmet); // set exit
// mode

document.TimeOutCtl.Show_STO_Msg(vcrh_ws_TO_show_msgTO); // disable
// display
// T.O. msg
document.TimeOutCtl.Set_STO_SKcmd(sz_str_redir_prebuf); // set
// string cmd
// for redir.

document.TimeOutCtl.Set_STO_Exit_Parms(sz_str_redir_parm); // string

document.TimeOutCtl.Start_STO()           // Start time computing ...

< /script >
```

The object will compute time elapsed. If there are mouse clicks events or keyboard pressed in the instance browser scope before the Time out programmed period - 30 minutes (in this case) the elapsed time will be cleared and there will continue computing the time. If there is no activity for a 30 minutes time period the application will be redirected to the programmed URL as selected.

See SAMPLE2 folder included in the distribution to get the source code sample.

8 - Control advanced features

8.1 - Debug

The control has the debug feature that can be enabled by making a directory entry in following path named as:

```
%UserEnviroment\crhwsto\crh_webTO_dbg.flg
```

Once the page the object is declared on, became instantiated, the object initialization tries to find that entry (usually use a file with length 0 bytes as a flag). If it finds the entry, the object becomes visible and the timer update is visible in the browser, so you can monitor the elapsed session time.

You also can enable this feature by applying the method **Set_STO_Debug**

Example:

```
<script LANGUAGE="JavaScript">
    // -----
    // set debug
    // -----
    var var_debug = 1
    var var_retval
    var_retval = document.TimeOutCtl.Set_STO_Debug ( var_debug )
```

```
if ( var_retval == 0) {  
    alert ( 'SessionTO Debug Enabled')  
} else{  
    alert ( 'SessionTO Debug Disabled')  
}  
</script>
```

8.2 - Log

The control has the log feature that can be enabled by making a directory entry named in following path:

%UserEnvironment\crhwsto\crh_webTO_log.flg

e.g. assuming the user is logged with the account USER01 you can enable log creating a file in following path:

win xp - C:\Documents and Settings\user01\crhwsto\crh_webTO_log.flg

win 7 - C:\Users\user01\crhwsto\crh_webTO_log.flg

Once the page where the object is declared is instantiated, the object initialization tried to find that entry (usually a file with 0 bytes) . If it finds the entry the object write sequentially all function call, as and events since its initialization , in following file :

%UserEnvironment\crhwsto\crh_webTO_log.txt

e.g. asuming the user is logged with the account USER01 you can find

win XP - C:\Documents and Settings\user01\crhwsto\crh_webTO_log.txt

win 7 - C:\Users\user01\crhwsto\crh_webTO_log.txt

This becomes useful at the time of time out control implementation stage, or to add more information in the Time Out exits tracking situation.

The log file structure looks like this (and speak from themselves) with the information organized as:

date / time / section involved / action

and the record is like:

2013/4/30 - 03:55:11 p.m. Web_ses_TO_Ctrl --- debug enabled ...

Following is a fragment of a **SessionTO** log file:

```
2013/4/26 - 06:14:51 p.m. Web_ses_TO_Ctrl --- log enabled...
2013/4/26 - 06:14:51 p.m. Web_ses_TO_Ctrl --- debug enabled...
2013/4/26 - 06:14:51 p.m. Web_ses_TO_Ctrl --- Initialize control NO SHM Found--
--
2013/4/30 - 03:55:11 p.m. Web_ses_TO_Ctrl --- debug enabled...
-----
2013/4/30 - 03:55:43 p.m. Set_STO_Debug -- Enable Debug...
2013/4/30 - 03:55:44 p.m. Init_STO - Session Timeout Inicializado - Timeout:
180 segundos
2013/4/30 - 03:55:45 p.m. Set_STO_Time - Timeout = 20 segundos
2013/4/30 - 03:56:04 p.m. ----- Keyboard hit ...
2013/4/30 - 03:56:13 p.m. ----- Mouse Click ...
2013/4/30 - 05:42:30 p.m. Web_ses_TO_Ctrl --- Init Register WinHand: 327846 --
-
2013/4/30 - 05:42:30 p.m. Set_STO_Time - Timeout = 20 segundos
2013/4/30 - 05:42:30 p.m. Set_STO_Url_2_Go = http://www.corhoma.com
2013/4/30 - 05:42:30 p.m. Start_STO - Refresh rate = 1
```

8.3 - Finish record

A control record is re/written every time the session ends by the Time Out condition in following path:

%UserEnvironment\crhwsto\crh_webTO_log.flg

e.g. assuming the user is logged with the account USER01 you can find

win XP - C:\Documents and Settings\user01\crhwsto\crh_webTO_END.dat

win 7 - C:\Users\user01\crhwsto\crh_webTO_END.dat

the record info the file crh_webTO_END.dat is as follows:

Crh_web_ses_TO - Time Out [20 s] en 2013/4/30 - 08:02:56 p.m.

where:

Data	Meaning
Crh_web_ses_TO	Control Name (constant)
Time Out [20 s]	programmed time out period
2013/4/30	Time Out date
08:02:56 p.m.	Time Out time

8.4 - Quantity close commands

The browser close action is performed by a automated key of one or several "ALT F4" keystrokes.

You can program the required quantity to exit from an application. The default quantity is 3 (three).

See the **Set_STO_Qsc** function in the control function section of this document.

8.5 Redirection parameters

If the selected exit method is redirection with arguments as :

Set_STO_Exit_Method(3) or **Set_STO_Exit_Method(2)**

it is possible to add arguments to be passed by GET METHOD to the redirected page.

This allows modify the operation of the page the control redirects to enable:

- CLEANUP
- AUDITS logs
- Special processes.
- etc

This parameters are to be found after three parameters the control force always every time this exit method is selected:

IP address; Workstation Name and Logged user

Example:

Assuming redirection to an URL named:

`http://secadmin/procedures/toexit.asp`

and the programmed option is:

ACCOUNT=BZA3759-12

the redirection URL received by the page will be like this:

http://secadmin/procedures/toexit.asp?IPWS=10.1.1.108&COMPNAME=CRHWKW705&USER=JABLASI&ACCOUNT=BZA3759-12

Then the page can capture all the parameters:

GET Variable	Meaning	Data
IPWS	workstatio IP adress	10.1.1.108
COMPNAME	workstation name	CRHWKW705
USER	logged user	JABLASI
ACCOUNT	selected as by implementor's needed in this example	BZA3759-12

“Text in red color means forced by the object”

9 - Function Index

Following listing organizes all the control functions by grouping them according to their role.

We distinguish **Control Functions** and **Operation Parameters**.

All the examples are written with javascript code (this is a browser object and we think the most common scripting for the browser enviroment is “Javascript”...)

9.1 - Control Functions

Name: Init_STO

Purpose: This function has to be called to initialize the control properly.

This function should be applied after apply an **exit_STO** function in order to reach a know state for control

Input: None

Output: an Integer with the value equal to 1

Example:

```
<script LANGUAGE="JavaScript">
    // -----
    // integer Init_STO()
    // -----
    var var_retval;
    var_retval = document.TimeOutCtl.Init_STO() ;
</script>
```

Name: Stop_STO

Purpose: This function stops the timer events and clears the elapsed time accumulator.

Input: None

Output: an integer with the value of elapsed time till the stop

Example:

```
<script LANGUAGE="JavaScript">
```



```
// -----  
// integer Stop_STO()  
// -----  
var var_retval ;  
var_retval = document.TimeOutCtl.Stop_STO() ;  
alert ( 'Time elapsed '+ var_retval) ;  
</script>
```

Name: ReStart_STO

Purpose: Re-Start time accounting after initialization

Input: None

Output: an integer with the value equal to 1

Example:

```
<script LANGUAGE="JavaScript">  
//-----  
// integer ReStart_STO()  
//-----  
var var_retval;  
var_retval = document.TimeOutCtl.ReStart_STO() ;  
alert ( 'Session Control Timer Restarted') ;  
</script>
```

Name: Show_STO_ver

Purpose: Shows the object software version

Input: None

Output: a String with the version info

Example:

Display the version for the installed control .

```
<script LANGUAGE="JavaScript">

    // -----
    //  String Show_STO_ver()
    // -----

    var var_version

    var_version = document.TimeOutCtl.Show_STO_ver()

    alert ( 'Time Out control version = ' + var_version )

</script>
```

Name: Start_STO

Purpose: Start time accounting after initialization of parameter

This function enable computing time till programed Time Out time period is reached. Remember that the time computing might be modified by the following functions :

Set_STO_NoRestime

Set_STO_Time

Input: None

Output: an integer with the value equal to 1

Example:

```
<script LANGUAGE="JavaScript">

    // -----

    // integer Start_STO()

    // -----
```

```
var var_retval ;  
var_retval = document.TimeOutCtl.Start_STO() ;  
alert ( 'Session Control Timer Started' ) ;  
</script>
```

Name: Exit_STO

Purpose: This function could be applied when it is desired to stop the timer and remove the mouse and keyboard event processing.

This action disable the session timer housekeeping with no possibility to recover It without reloading the control and re-program it.

Input: None

Output: None

Example:

```
<script LANGUAGE="JavaScript">  
    // -----  
    // Exit_STO()  
    // -----  
    document.TimeOutCtl.Exit_STO() ;  
    alert ( 'Timer stop and keyb / mouse events capture removed ' );  
</script>
```

Name: Clear_STO

Purpose: This function set the elapsed time accumulator to zero without stopping the timer .

Input: None

Output: an integer with the value of elapsed time till the action of clearing the time accumulator.

Example:

```
<script LANGUAGE="JavaScript">
    // -----
    // integer Clear_STO()
    // -----
    var_retval;
    var_retval = document.TimeOutCtl.Clear_STO() ;
    alert ( 'Session Control Timer cleared');
</script>
```

9.2 - Operation Parameters Functions

Name: Set_STO_Time

Purpose: This function set the Time Out time period.

Input: Integer between 0 and 3600 meaning the session duration
in seconds

If **argument < 0** - error and it will be set to default
and time out will not set enabled

If **argument > 3600** - it will be set to 3600
and time out will be set enabled

If **argument = 0** - it will be set to default period (30s)
and time out will be set enabled

If **0 < argument < 3600** - it will be set to argument value
and time out will be set enabled

Output: Integer with the Time Out period value

Example

In this example session time period will be set up to 300s (5 minutes)

```
<script LANGUAGE="JavaScript">
//-----
// integer Set_STO_Time(t As Integer)
//-----
var vcrh_ws_TO_time = 300s ;           // duracion sesión en segundos
document.TimeOutCtl.Set_STO_Time( vcrh_ws_TO_time ); // Set time period
</script>
```

Name: Set_STO_Url_2_Go

Purpose: This function sets the page where the browser redirects the operation

if the exit method selected was 2 or 3 and the time out was enabled.

Input: String describing the URL where the redirection should be done

once the time out period has reached and time out is enabled

If the string is null the URL will point to the object default URL :

http://www.corhoma.com/prod/session_to/sample_exit.php

Output: an Integer with the value equal to 1

Example

```
<script LANGUAGE="JavaScript">
// -----
// integer et_STO_Url_2_Go(url As String)
// -----
var vcrh_app_exitmet = 2 ;
var vcrh_ws_TO_url2go =http://www.corhoma.com/prod/session_to/sample_exit.php";
document.TimeOutCtl.Set_STO_Url_2_Go(vcrh_ws_TO_url2go);    // Set URL
document.TimeOutCtl.Set_STO_Exit_Method(vcrh_app_exitmet);
</script>
```

Name: Set_STO_Exit_Method

Purpose: This function selects the control behavior after reaching the time out condition (whenever the T.O. was enabled).

Input: Integer with the value of 1, 2 or 3

It can be selected three exit methods :

- 1 - Close the **IEExplore** at the Time Out condition

2 - Redirect a page at the Time Out condition

3 - Redirect but from a modal dialog window

Output: an Integer value with the same input argument received as a parameter.

Example

In this example , when the T.O. period is reached the application close the modal dialogs and redirect to a exit page named clf.php

```
<script LANGUAGE="JavaScript">
// -----
// integer Set_STO_Exit_Method(em As Integer)
// -----
var vcrh_app_exitmet = 3 ; //
var sz_str_redir_prebuf = "{F11}%D{DEL}"; // comand PONT MOUSE TO
// ADDRESS BAR
var vcrh_ws_TO_url2go = "http://vmcmfiquaapp:6050/sw3admc/clf.php";

// Set URL ifi Time Out
document.TimeOutCtl.Set_STO_Url_2_Go(vcrh_ws_TO_url2go);

// Set the exit method
document.TimeOutCtl.Set_STO_Exit_Method(vcrh_app_exitmet);
</script>
```

Name: Set_STO_Exit_Parms

Purpose: This function set the parameters string will be sent beside the redirection URL so the browser redirect the page to a known one passing parameters to that page with GET method .
By default the object always add following values to be passed by GET method in that order:

Input: a String - Could be a null string a. The maximum length is **120** char.

This string should begin with "?" character.

Output: Integer - Zero means OK and not zero means error

Example

In this example, at the time out condition the page will be redirected to other administration site by passing a required info .

```
<script LANGUAGE="JavaScript">
// -----
// integer Set_STO_Exit_Parms(ep As String)
// -----
var vcrh_ws_TO_url2go = "http://admin:8090/endbyto/procend.php"; // where
var sz_str_redir_parm = "?OPCODE=<?=$ctrl_value ?>" // required parameter for
// an hypothetical app.

var var_aux ;
var_aux = document.TimeoutCtl.Set_STO_Url_2_Go(vcrh_ws_TO_url2go);
var_aux = document.TimeoutCtl.Set_STO_Exit_Parms(sz_str_redir_parm); //
if ( var_aux == 0 ) {
    alert ('It will be a redirected to the defined T.O. procedure politic ');
}
```



```
}  
</script>
```

Name: Show_STO_Msg

Purpose: Enable / disable show T.O. message

Input: - Integer - Value of zero disable message T.O. (default)

Value of 1 enable message T.O.

Output: Integer - the same value as argument (0 or 1) with the same meaning that input parm

Example

In this example there will be a message in french when the T.O. is reached

```
<script LANGUAGE="JavaScript">  
// -----  
// integer Show_STO_Msg(t As Integer)  
// -----  
var var_aux ;  
var vcrh_ws_TO_msg = "La session a pris fin"; //  
var var_dspTomsg = 1  
var_aux ; document.TimeOutCtl.Show_STO_Msg(var_dspTomsg)  
var_aux ; document.TimeOutCtl.Set_STO_To_msg(vcrh_ws_TO_msg);  
</script>
```

Name: Set_STO_TO_msg

Purpose: Set thge message to be displayed (if enabled)
as the Time Out message when the T.O. period is reached.
Otherwise the message will be the default one.

Input: String - defining the message to be displayed

Output: an Integer with value 1

Example

```
<script LANGUAGE="JavaScript">  
// -----  
// integer Set_STO_TO_msg(msg1 As String)  
// -----  
  
var vcrh_ws_TO_msg = "This is the message to show with the T.O. condition "; //  
  
var var_aux ;  
  
var var_dspTomsg = 1  
  
var_aux = document.TimeOutCtl.Show_STO_Msg(var_dspTomsg)  
var_aux = document.TimeOutCtl.Set_STO_To_msg(vcrh_ws_TO_msg);  
</script>
```

Name: Set_STO_Qsc

Purpose: Set the quantity of exit comand to send
when the exit method selected was
and the time Out period was reached

Input: Integer - Should be a value greater than cero and lower than 6.
If argument value in lower than zero the comands quantity of is
set to 3 (default value) ,

If argument value is greater than 6 the command quantity of is set to 6, and the control will display a message box suggesting the value looks too big

Output: an Integer - the return value is the quantity set by the function .

Example

In this example the command quantity of will be set to 3

```
<script LANGUAGE="JavaScript">
// -----
// integer Set_STO_Qsc(n1 As Integer)
// -----
var var_aux ;
var var_qty = -7;

// set refresh timer eac 5 second
var_aux = document.TimeOutCtl.Set_STO_Qsc(var_qty);
alert ('Quantity of send exit command set to = ' + var_aux ) ;
</script>
```

Name: Set_STO_TrfsH

Purpose: Start time accounting after initialization

Input: This parameter define in seconds the refresh rate for timer accumulator.(i.e. the minimal time resolution to

get for the Time Out). If the session period is enough large, setting this period to a number greater than the default one can improve the browser performance (if might be degraded)
If time period **is negative** functions sets the default value.
If time period **is greater than** maximum allowable refresh period the period time is set a s the maximum allowable

Output: Integer - return the argument value if it is not negative or greater than maxium alloable refresh period.

Example

In this example the refresh time is set to 5 seconds.

```
<script LANGUAGE="JavaScript">  
  // -----  
  // integer Set_STO_Trsh(t1 As Integer)  
  // -----  
  var var_aux ;  
  var var_refsh = 5;  
  
  // set refresh timer each 5 second  
  var_aux = document.TimeOutCtl.Set_STO_Trsh(var_refsh);  
  
  alert ('Time refrist period = ' + var_aux ) ;  
</script>
```

Name: Set_STO_Debug

Purpose: Start time accounting after initialization

Input: None

Output: an Integer

Example

This example shows a function to toggle the control debug mode.

```
<script LANGUAGE="JavaScript">
// -----
// integer Set_STO_Debug(d As Integer)
//-----

var var_aux ;
var_exmp_debug = 0 ;
function toggle_debug()    {
    if ( vcrh_ws_TO_dbg == 0 ) {
        vcrh_ws_TO_dbg = 1 ;
        alert ('Debug Enabled') ;
    } else {
        vcrh_ws_TO_dbg = 0 ;
        alert ('Debug Disabled') ;
    }
}
}

var_aux = document.TimeOutCtl.Set_STO_Debug(var_exmp_debug);
```

</script>

Name: Set_STO_SKcmd

Purpose: This function set the keystrokes buffer to send to the browser
if exit method is 2 o 3 and if the Time Out condition was reached.

Input: a String describing keystrokes to handle IEXPLORE

- {F6} set focus on IEXPLORE address bar
- {^O} Open file - Open dialog box to handle file selection
- {^T} Open a new browser tag
- {^N} - OPen a private window

Output: an Integer - 1 means error - string greater than MAX_prebuff_len
0 means function performed OK

e.g. Keystrokes to handle IEXPLORE

String	Action
{F11}	Set full screen
{%D}	Set focus on IEXPLORE address bar
{^O}	Open file - Open dialogbox to handle file selection
{^T}	Open a new browser tag
{^N}	Open a private window

See the **Appendix 2** for more details on this topic

Example

In this example if redirection is the exit mode; when occurs the time out event the IEXplorer containing the objet, goes full screen, set focus in address bar and delete characters in the explorer address bar then being directed to where **var_where2go** refers .

```
<script LANGUAGE="JavaScript">
// -----
// integer Set_STO_SKcmd(sz As String)
// -----
var var_argx = ""{F11}%D{DEL}" // if exit mode is redirection
var var_retval
var var_where2go = "http://sales01svr:8080/operations/toexit.php";

document.TimeOutCtl.Set_STO_Url_2_Go(vcrh_ws_TO_url2go);
var_retval = document.TimeOutCtl.Set_STO_SKcmd ( var_argx ) ;

</script>
```

Name: Set_STO_NoResTime

Purpose: Disable / Enable resetting computing elapsed time

Input: an Integer

0 means **Reset** elapsed time accumulator each mouse /
keyboard event in the related windows

<> 0 means **NO RESET** elapsed time each mouse / keyboard event in
the related windows

Output: an Integer with same meaning as the Input value

Example

In this example the time accumulator will be reset with each related window Mouse / Keyboard event

To evaluate it modify the var_argx to not equal to zero.

```
<script LANGUAGE="JavaScript">
    // -----
    // integer Set_STO_NoResTime(b As Integer)
    //-----
    var var_argx = 0
    var var_retval

    var_retval = document.TimeOutCtl.Set_STO_NoResTime ( var_argx ) ;

    if ( var_retval == 0 ) {
        alert ( 'Elapsed time will be reset each related window Mouse/Keyb event ' )
    } else{
alert ( ' Elapsed time will not be reset each window //
        Mouse /Keyboard event no matter where were them')
    }

</script>
```


Appendix I

Object and functions default values

Clear_STO : – nothing initially to be defined for this

Init_STO -

Set_STO_Debug - initially set to 0 - disable

Set_STO_Exit_Method – set to 1 – (close browser)

Set_STO_Exit_Parms – Empty

Set_STO_NoResTime – reset accumulator time with keystroke / mouse-click

Set_STO_Qsc - quantity equal to 3

Set_STO_Time – session time set to 30 seconds

Set_STO_TO_msg –

Set_STO_TrfsH _ - initially set to one second

Set_STO_Url_2_Go – initially set to :

http://www.corhoma.com/prod/session_to/sample_exit.php

Show_STO_Msg – initially set disable

Show_STO_ver – nothing initially to be defined for this

Start_STO – None – nothing is initially to be defined for this

Appendix II

Keystrokes Codification for Set_STO_SKcmd function

You can send keys to the active window, as if the user was pressing keys on the keyboard by using the Send Keystrokes programmatically, This is the method the control uses to control the IEXPLORE once the timeout is reached and you have to do something to conclude / act

Following is a list of keystroke key codes

Key	Code
BACKSPACE	{BACKSPACE}, {BS}, or {BKSP}
BREAK	{BREAK}
CAPS LOCK	{CAPSLOCK}
DEL or DELETE	{DELETE} or {DEL}
DOWN ARROW	{DOWN}
END	{END}
ENTER	{ENTER}or ~
ESC	{ESC}
HELP	{HELP}

Key	Code
LEFT ARROW	{LEFT}
TAB	{TAB}
HOME	{HOME}
INS or INSERT	{INSERT} or {INS}
NUM LOCK	{NUMLOCK}
PAGE DOWN	{PGDN}
PAGE UP	{PGUP}
PRINT SCREEN	{PRTSC}
RIGHT ARROW	{RIGHT}
SCROLL LOCK	{SCROLLLOCK}
UP ARROW	{UP}
F1	{F1}
F2	{F2}
F3	{F3}
F4	{F4}
F5	{F5}
F6	{F6}
F7	{F7}
F8	{F8}
F9	{F9}
F10	{F10}
F11	{F11}
F12	{F12}

Key	Code
F13	{F13}
F14	{F14}
F15	{F15}
F16	{F16}

To specify keys combined with any combination of the SHIFT, CTRL, and ALT keys, precede the key code with one or more of the following codes:

Key	Code
SHIFT	+
CTRL	^
ALT	%

To specify that any combination of SHIFT, CTRL, and ALT should be held down while several other keys are pressed, enclose the code for those keys in parentheses.

For example, to specify to hold down SHIFT while E and C are pressed, use "+(EC)".

To specify to hold down SHIFT while E is pressed, followed by C without SHIFT, use "+EC".

To specify repeating keys, use the form {key number}. You must put a space between key and number.

For example, {LEFT 36} means press the LEFT ARROW key 36 times; {H 15} means press H 15 times.

Appendix III

Sample exit page

Following is the default demo site code that processes the control redirection

It is located in:

http://www.corhoma.com/prod/session_to/sample_exit.php

Here is the code (a php sample coded)

```
<html>
<head>
  <title>Session_TO Exit redirection exit page demo</title>
</head>
<body>

<?php
echo "<br><br>";
echo "-----<br>" ;
echo " SessionTO exit demo page - (c) Corhoma SRL - www.corhoma.com <br>";
echo "<br>";
echo " visit SessionTO site @ www.sessionTO.com <br>" ;
echo "-----<br>" ;
$var1= $_GET["IPWS"] ;
$var2= $_GET["COMPNAME"] ;
$var3= $_GET["USER"] ;
echo " <br><br>IP adress = ". $var1 ;
echo " <br><br>Workstation = ". $var2 ;
```

```
echo " <br><br>User = ". $var3 ;
$ffdd = 'Y M d H:i:s' ;
$fecha= date ($ffdd) ;
echo " <br><br>Date = ". $fecha ;
$data = date ($ffdd)." ".$var1 ." ".$var2 ." ".$var3 ;
$fh = fopen("log.txt", "a+");
    if($fh==false)
        die("cant create file");
    fwrite($fh, $data);
flush($fh) ;
fclose($fh);
echo " <br><br> ";
echo " <br> -----" ;
echo " <br> You have been redirected from a page with Sessio_TO control demo
version ..." ;
echo " <br> -----" ;
echo " <br><br> ";
?>
```