



Sierra Engine

Applications development for Psion Teklogix
radio-frequency terminals with Sierra Engine

Reference

SANTA  FE TECH

© 2003 Santa Fe Tech Software Development. All rights reserved.

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Santa Fe Tech Software Development on the issues discussed as of the date of publication.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Santa Fe Tech.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Teklogix, TSDK Software Development Kit, and WinTSF are trademarks of Teklogix Inc., Mississauga, Ontario, Canada.

Santa Fe Tech Software Development • Buenos Aires, Argentina.

Introduction

The present document is a technical reference. It is directed to the programmers that want to use Sierra Engine to develop radio frequency applications with Psion Teklogix terminals.

It supplements to the **Programmer's Guide**.

It contains a description of each property, method and public event defined by the class **cSE_Main**.

The class **cSE_Main** is the only exposed class of the dynamic link library **SierraEngine.dll** (Activex DLL)

Several code examples in Visual Basic™ are also included.

Clase cSE_Main

Events

SEConfigRequest	SEFirstPageRequired
SEEndApplication	SEKeyPress
SEEndSession	

Properties

SE_AppButtonIcon	SE_Page
SE_AppLogo	SE_TerminalID
SE_AppTitle	SE_TSEFileName
SE_FieldData	SE_UserLogged
SE_MaxTerminalsUsed	

Methods

SE_Bell	SE_StartEngine
SE_ClearAllEntryFields	SE_StartEntry
SE_EndEngine	
SE_FmtTerm	
SE_HeyYouMessage	
SE_Log	
SE_ResetTerminalStatus	

SE_AppButtonIcon Property

Description Get or set the icon for the toolbar button reserved for the client application.

Syntax [*SE_object*.]SE_AppButtonIcon [= *string*]

Remarks The SE_AppButtonIcon property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main

The value that it receives must be a string that specifies a file name, it can include folder and unit, and it must correspond to an image.

The recognized formats are: bitmap (.bmp), icon (.ico), run-length encoded (.rle), metafile (.wmf), enhanced metafiles (.emf), GIF (.gif) y JPEG (.jpg).

If no value is set to this property, the toolbar button reserved to the application will not be displayed.

Example

(see example in SE_AppTitle)

SE_AppLogo Property

Description Get or set the application logo displayed in the main window.

Syntax [*SE_object*.]SE_AppLogo [= *string*]

Remarks The SE_AppLogo property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main

The value that it receives must be a string that specifies a file name, it can include folder and unit, and it must correspond to an image.

The recognized formats are: bitmap (.bmp), icon (.ico), run-length encoded (.rle), metafile (.wmf), enhanced metafiles (.emf), GIF (.gif) y JPEG (.jpg).

If no logo is defined, the default value for this property is the Santa Fe Tech logo.

Example

(see example in SE_AppTitle)

SE_AppTitle Property

Description Get or set the application title beginnings with "SE-".

Syntax [*SE_object*.]SE_AppTitle

Remarks The SE_AppTitle property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main

The title is shown in the upper bar of the Sierra Engine main window (caption).

If no title is defined, the default value for this property is the "Sierra-Engine" title.

Example

The example shows the use of SE_AppTitle, SE_AppLogo and SE_AppButtonIcon properties, to personalize the application. They are used inside the Form_Load event.

Here, an instance of cSE_Main is created with the name oSE_Main.

The title, the logo and the button icon are defined using the properties SE_AppTitle, SE_AppLogo and SE_AppButtonIcon respectively.

The name and path of the .tse file are set using the SE_TSEFilename property.

The name and path of the log file are set using the SE_LogPathFileName property. The Sierra Engine is started invoking SE_StartEngine.

```
Private Sub Form_Load()
    TSE_FILENAME = App.EXENAME & ".tse"
    Set oSE_Main = New cSE_Main
    Set oSE = oSE_Main
    oSE_Main.SE_AppTitle = App.Comments
    oSE_Main.SE_AppLogo = App.Path & "\media\" & "ApplicationLogo.jpg"
    oSE_Main.SE_AppButtonIcon = App.Path & "\media\" &
"ApplicationButtonIcon.ico"
    oSE_Main.SE_TSEFileName = App.Path & "\tse\" & TSE_FILENAME
    oSE_Main.SE_LogPathFileName = App.Path & "\" & App.EXENAME & ".txt"
    oSE_Main.SE_StartEngine
End Sub
```

SE_Bell Method

Description Sound the terminal beeper.

Syntax [*SE_object*.]SE_Bell [*terminal_number*]

Remarks The SE_Bell method has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal. If terminal number is 0 and the virtual terminal is in use sounds a tone through the computer's internal speaker.

Example

The example uses SE_Bell to sound a tone if the user presses a function key higher than F5 (i.e. F6 thru F32)

This method is called inside the event SEKeyPress.

Here oSE_Main is the instance of cSE_Main used by the application and Page_Login, and Page_menu are procedures for the selected pages "Login" and "Menu" respectively.

```
Private Sub oSE_Main_SEKeyPress(terminal_Number As Byte, _
                               page_Name As String, key as Key_Enum)

    If key > 5 Then
        oSE_Main.SE_Bell terminal_Number
    End If
    oSE_Main.SE_StartEntry terminal_Number
    Exit Sub
    Select Case page_Name
        Case "Login"
            Page_Login terminal_Number, key
        Case "Menu"
            Page_Menu terminal_Number, key
    End Select
End Sub
```


SE_ClearAllEntryFields Method

Description Clear all entry fields on the selected page.

Syntax [*SE_object*.]SE_ClearAllEntryFields [*terminal_number*]

Remarks The SE_ClearAllEntryFields method has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

The page selected is the page received as parameter in the SEKeyPress event or the page assigned in a call to the SE_Page property.

Example

The example uses SE_ClearAllEntryFields to ensure that all entry fields are empty when the "Login" page is displayed. This method is called inside the event SEFirstPageRequired.

Here oSE_Main is the instance of cSE_Main used by the application. The field "Radio#.f" is filled with the physical number of the terminal (also known as radio ID) and the cursor is positioned on the entry field "Username.e" and the keyboard is unlocked.

```
Private Sub oSE_Main_SEFirstPageRequired(terminal_Number As Byte)
    oSE_Main.SE_Page(terminal_Number) = "Login"
    oSE_Main.SE_ClearAllEntryFields terminal_Number
    oSE_Main.SE_FieldData(terminal_Number, "Radio#.f") = _
        oSE_Main.SE_TerminalID(terminal_Number)
    oSE_Main.SE_StartEntry terminal_Number, "Username.e"
End Sub
```

SEConfigRequest Event

Description This event is fired when the user clicks on the toolbar button reserved to the client application.

Syntax Sub [*SE_object_*]SEConfigRequest(*Mode*)

Remarks The SEConfigRequest event has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>Mode</i>	Enum. Determines if the Sierra Engine is mode normal or the user has entered in Setup mode by pressing F1 on the start-up

The possible values for *Mode* are:

MODE_NORMAL = 1

MODE_CONFIG = -1 (setup mode)

This event can be used to invoke a configuration dialog for the application when in MODE_CONFIG and to open any other menu when in MODE_NORMAL, for example to select and print reports.

Remember to keep the system load as minimum as possible when in normal mode to don't degrade performance.

SEEndApplication Event

Description This event is fired when the user closes the Sierra Engine main windows and terminates the application or when the Sierra Engines detects and unrecoverable error.

Syntax Sub [*SE_object_*]SEEndApplication(*error_description*)

Remarks The SEEndApplication event has the following parts:

<u>Part</u>	<u>Description</u>
<i>SE_object</i>	Instance of cSE_Main
<i>error_description</i>	String. Is the description of the error that fired the SEEndApplication event. If the event is fired by the user closing the Sierra Engine main window <i>error_description</i> is empty.

Use this event to release *SE_Object*. No other Sierra Engine event is fired after that one. All the terminals sessions are closed and all resources released.

Example

```
Private Sub oSE_Main_SEEndApplication(error_description As String)
    Set oSE_Main = Nothing
End
End Sub
```

SE_EndEngine Method

Description Closes all terminals sessions and all Sierra Engine windows, then fire the SEEndApplication event

Syntax [*SE_object*.]SE_EndEngine

Remarks The SE_EndEngine method has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main

Use this method to end the Sierra Engine operation from your application. Once Sierra Engine closes all terminals sessions and releases all resources, the windows are closed and the SEEndApplication event is fired. No SEEndSession event is fired.

SEEndSession Event

Description This event is fired in response to the SE_ResetTerminalStatus method or when a terminal is locked using Terminal Status dialog when in normal mode.

Syntax Sub [*SE_object_*]SEEndSession(*terminal_number*)

Remarks The SEEndSession event has the following parts:

<u>Part</u>	<u>Description</u>
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

When in normal mode (the other possible mode is config or “setup” mode, when the user press F1 on start-up) Sierra Engine allows the user to lock/unlock any terminal using the Terminal Status dialog. When a terminal is locked the session is closed. And this event is fired so your application can, for example, set the terminal user previously logged on this terminal as not logged.

Also this event is fired when the SE_ResetTerminalStatus method is called.

SE_FieldData Property

Description	Get or set field contents.
Syntax	[<i>SE_object_</i>]SE_FieldData(<i>terminal_number</i> , <i>field_name</i>) [= <i>string</i>]

Remarks The SE_FieldData property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.
<i>field_name</i>	String. Name of the field to get or set contents

The max. field data length is 80 characters. If the field does not exist in the selected page SE_FieldData, returns an empty string

Example

The example uses SE_FieldData to fill the field "Radio#.f" with the physical number of the terminal (also known as radio ID). This property is used inside the event SEFirstPageRequired.

Here oSE_Main is the instance of cSE_Main used by the application. The cursor is positioned on the entry field "Username.e" and the keyboard is unlocked.

```
Private Sub oSE_Main_SEFirstPageRequired(terminal_Number As Byte)
    oSE_Main.SE_Page(terminal_Number) = "Login"
    oSE_Main.SE_ClearAllEntryFields terminal_Number
    oSE_Main.SE_FieldData(terminal_Number, "Radio#.f") = _
        oSE_Main.SE_TerminalID(terminal_Number)
    oSE_Main.SE_StartEntry terminal_Number, "Username.e"
End Sub
```

SEFirstPageRequired Event

Description	This event is fired when a terminal is attached to the application.
Syntax	Sub [<i>SE_object</i>]SEFirsPageRequired(<i>terminal_number</i>)
Remarks	The SEFirstPageRequired event has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

When Sierra Engine begins in normal mode (see SE_StartEngine) all terminals defined as not locked are asked for acknowledge. For every terminal that is ready to connect a SEFirstPageRequired event is fired because Sierra Engine needs to know which page or screen defined in the .tse file is the first page to send to the terminals (see example below) This event is also fired when a terminal locked is unlocked using the Terminal Status dialog.

Example

In this example inside the SEFirstPageRequired event a call to SE_Page tells Sierra Engine to display the page "Login" as the first page every time a terminal is connected.

Here oSE_Main is the instance of cSE_Main used by the application.

The example uses SE_FieldData to fill the field "Radio#.f" with the physical number of the terminal (also known as radio ID). The cursor is positioned on the entry field "Username.e" and the keyboard is unlocked.

```
Private Sub oSE_Main_SEFirstPageRequired(terminal_Number As Byte)
    oSE_Main.SE_Page(terminal_Number) = "Login"
    oSE_Main.SE_ClearAllEntryFields terminal_Number
    oSE_Main.SE_FieldData(terminal_Number, "Radio#.f") = _
        oSE_Main.SE_TerminalID(terminal_Number)
    oSE_Main.SE_StartEntry terminal_Number, "Username.e"
End Sub
```

SE_FmtTerm Function

Description This function returns the string passed as argument formatted to fit the terminal width.

Syntax `[SE_object_]SE_FmtTerm(terminal_number, message, {alignment})`

Remarks The SE_FmtTerm function has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.
<i>Message</i>	String. The message to format.
<i>Alignment</i>	Optional. Enum. The horizontal alignment of the message on the terminal display.

The possible values for *alignment* are:

ALIGN_CENTER = 0 (default)
 ALIGN_RIGHT = 1
 ALIGN_LEFT = -1

Returns a string. If the text is wider than the terminal display it will be truncated. Especially useful when need to center messages.

SE_HeyYouMessage Method

Description The message `message` is displayed on the bottom line of the screen and triggers a beep.

Syntax `[SE_object.]SE_HeyYouMessage(terminal_number, message)`

Remarks The `SE_HeyYouMessage` method has the following parts:

Part	Description
<i>SE_object</i>	Instance of <code>cSE_Main</code>
<i>terminal_number</i>	Byte. The logical number of the terminal.
<i>Message</i>	String. The message to display.

After the message is displayed the terminal keyboard is unlocked. The operator may press <ENTER> and the message is cleared. The message is centered, if exceeds the display width is truncated.

Example

In this example the length of the text inside the entry field "new_password" is checked to have 5 characters. If not, a `SE_HeyYouMessage` is sent to the terminal. Here `oSE_Main` is the instance of `cSE_Main` used by the application.

```
if Len(oSE_Main.FieldData(terminal_Number, "new_password")) <> 5 then
    oSE_Main.SE_HeyYouMessage terminal_Number, _
                                "Password must be 5 characters long"
end if
```

SEKeyPress Event

Description This event is fired when <enter> or a function key is pressed by the terminal user.

Syntax Sub [*SE_object_*]SEKeyPress(*terminal_number*, *page_name*, *key*)

Remarks The SEKeyPress event has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.
<i>page_name</i>	String. The name of the selected page
<i>key</i>	Enum. The key pressed.

The possible values for *key* are:

```
KEY_HEY_YOU = -1
KEY_ENTER = 0
KEY_F1 = 1
KEY_F2 = 2
....
KEY_F34 = 34
```

When the terminal keyboard is unlocked for data entry, the terminal works independently from the host computer until the terminal operator indicates that all the data required for the displayed page has been entered. The terminal then locks the keyboard, and Sierra Engine fires a SE_KeyPress event with the terminal number, the name of the page displayed and the key pressed.

This event will be fired when one of the following occurs:

- An operator enters data in a transmit-on-entry field.
- An operator presses a function key or <enter>.
- An operator responds to a hey-you message.

SEKeyPress Event (Cont.)

When operator enters data in a transmit-on-entry field, *key* value is set to KEY_ENTER.

The number of function keys available depends on the terminal model used.

When operator responds to a hey-you message, pressing enter *key* value is set to KEY_HEY_YOU

Example

In this example we have different procedures for each page. Here oSE_Main is the instance of cSE_Main used by the application.

The procedure for page "menu" is shown below:

```
Private Sub oSE_Main_SEKeyPress(terminal_Number As Byte, _
                               page_Name As String, key As Key_Enum)
    Select Case page_Name
        Case "logo"
            ProdPageLogo terminal_Number, key
        Case "menu"
            ProdPageMenu terminal_Number, key
        Case "info"
            ProdPageInfo terminal_Number, key
        Case "passwd"
            ProdPagePasswd terminal_Number, key
    End Select
End Sub

Private Sub ProdPageMenu(ByVal terminal_Number As Byte, _
                        ByVal key As Key_Enum)
    'Process messages from page "menu".
    If key = KEY_F1 Then
        oSE_Main.SE_Page(terminal_Number) = "info"
        oSE_Main.SE_StartEntry terminal_Number, "code.e"
    ElseIf key = KEY_F2 Then
        oSE_Main.SE_Page(terminal_Number) = "passwd"
        oSE_Main.SE_StartEntry terminal_Number, "oldpasswd.e"
    ElseIf key = KEY_F5 Then
        oSE_Main.SE_Page(terminal_Number) = "logo"
        oSE_Main.SE_StartEntry terminal_Number
    Else
        ' unlock the keyboard for any other function key
        oSE_Main.SE_StartEntry terminal_Number
    End If
End Sub
```

SE_Log Method

Description	Writes to the log.
Syntax	[<i>SE_object</i> .]SE_Log(<i>terminal_number</i> , <i>icon_number</i> , <i>description</i>)
Remarks	The SE_Log method has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.
<i>icon_number</i>	Enum. Icon to be displayed on log.
<i>Description</i>	String. Description to be displayed on log.

The possible values for *icon_number* are:

```

ICON_SYSTEM = 1
ICON_WARNING = 2
ICON_RED_TERM = 3
ICON_YELLOW_TERM = 4
ICON_GREEN_TERM = 5
ICON_RED_MAN = 6
ICON_YELLOW_MAN = 7
ICON_GREEN_MAN = 8
ICON_BLACK_MAN = 9

```

Sierra Engines writes to the log the following events:

- The Sierra Engine starts.
- A terminal is connected, before fire an SEFirstPageRequired event.
- The method SE_UserLogged is called.
- The method SE_ResetTerminalStatus called.
- The virtual terminal window is opened.
- The virtual terminal window is closed.
- A change in the configuration is made.
- The status of any terminal is changed by the user.
- A message is sent to a terminal from the Terminal Status Dialog.
- An error is detected.

Also, using this method the application can write to the log file.
The log file is a plain text file delimited with tabs (ASCII 09).
The log file is located on the application path and is named Sierra_Log.txt

The last 10 log entries are shown in the Sierra Engine main windows.

Every log entry has the following fields:

Type, Date, Time, Terminal, user_Name, and Description

- *Type* is the icon number.
- *Date* and *Time* are filled with the actual values of date and time.
- *Terminal* is the **physical** terminal number (radio ID).
- *user_Name* is the user logged to this terminal, if no user is logged or the user name was not set by SE_UserLogged property, *user_Name* is blank.
- *Description* is the parameter *description* in the SE_Log call.

Example

In this example the page "SelectWarehouse" is selected and displayed, the keyboard is unlocked and an entry to the log is made with the legend "On page 'Select Warehouse'". Here oSE_Main is the instance of cSE_Main used by the application.

```
oSE_Main.SE_Page(terminal_Number) = "SelectWarehouse"  
oSE_Main.SE_StartEntry terminal_Number  
oSE_Main.SE_Log terminal_Number, ICON_BLACK_MAN, "On page  
'SelectWarehouse'"
```

SE_MaxTerminalsUsed Property

Description Read only. Get the number of terminals defined in configuration mode to be used by the application.

Syntax [*SE_object_*]SE_MaxTerminalsUsed

Remarks The SE_MaxTerminalsUsed property has the following parts:

<u>Part</u>	<u>Description</u>
<i>SE_object</i>	Instance of cSE_Main

The number of terminals to use must be equal or less than the number of terminals licensed in Sierra Engine.

If you have licensed for 10 terminals but you will no use more than 8 terminals, is better to set SE_MaxTerminalsUsed equal 8 rather than equal 10, because Sierra Engine allocates some resources for every terminal to SE_MaxTerminalsUsed, even if the terminal is not connected.

SE_Page Property

Description Get or set the selected page.

Syntax [*SE_object*]SE_Page(*terminal_number*) [= *string*]

Remarks The SE_Page property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

When you assign a page name to the SE_Page property the page became the selected page and is displayed. Further calls to SE_FieldData, SE_ClearAllEntryFields, and SE_StartEntry will apply to this page.

Example

In this example inside the SEFirstPageRequired event a call to SE_Page tells Sierra Engine to display the page "Login" as the first page every time a terminal is connected

Here oSE_Main is the instance of cSE_Main used by the application.

The example uses SE_FieldData to fill the field "Radio#.f" with the physical number of the terminal (also known as radio ID). The cursor is positioned on the entry field "Username.e" and the keyboard is unlocked.

```
Private Sub oSE_Main_SEFirstPageRequired(terminal_Number As Byte)
    oSE_Main.SE_Page(terminal_Number) = "Login"
    oSE_Main.SE_ClearAllEntryFields terminal_Number
    oSE_Main.SE_FieldData(terminal_Number, "Radio#.f") = _
        oSE_Main.SE_TerminalID(terminal_Number)
    oSE_Main.SE_StartEntry terminal_Number, "Username.e"
End Sub
```

SE_ResetTerminalStatus Method

Description Clears User, Date, and Time on Terminal Status dialog. Fires SEEndSession event.

Syntax [*SE_object*.]SE_ResetTerminalStatus(*terminal_number*)

Remarks The SE_ResetTerminalStatus method has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

When calling this method Sierra Engine does the following:

Clears the fields User, Date, and Time on the Terminal Status Dialog and set the status to "Connected".

Sends a "User not logged" to the log.

Fires a SEEndSession event.

Usually you call this method when return to the logon page.

Example

In this example the "Login" page is selected and displayed, all entry fields are cleared (i.e. user and password), and the cursor is moved to the "Username.e" field. Then a call to SE_ResetTerminalStatus prepares to the a new login. Here oSE_Main is the instance of cSE_Main used by the application.

```
oSE_Main.SE_Page(terminal_Number) = "Login"
oSE_Main.SE_ClearAllEntryFields terminal_Number
oSE_Main.SE_StartEntry terminal_Number, "Username.e" ' unlock keyboard
oSE_Main.SE_ResetTerminalStatus terminal_Number
```


SE_StartEngine Method

Description Shows the Sierra Engine main windows. Begin process.

Syntax [*SE_object*.]SE_StartEngine

Remarks The SE_StartEngine method has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main

After this method, a call to SE_TSEFilename property must be made, setting the .tse filename to use.

SE_StartEngine starts the log record, shows the main window and prompts for F1 key to enter the setup mode. After 3 seconds, if F1 is not pressed, starts the normal node. The link with the RF base station is started, reads the .tse file and the pooling for terminals messages begins.

Example

In this example an instance of cSE_Main is created with the name oSE_Main. The name and path of the .tse file is set using the property SE_TSEFilename and the Sierra Engine is started.

```
Set oSE_Main = New cSE_Main
oSE_Main.SE_TSEFilename = App.Path & "\myTSEfile.tse"
oSE_Main.SE_StartEngine
```

SE_StartEntry Method

Description	Unlocks the terminal keyboard. Moves the cursor to the entry field.
Syntax	<code>[SE_object.]SE_StartEntry(<i>terminal_number</i>, {<i>field_name</i>})</code>
Remarks	The SE_StartEntry method has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.
<i>field_name</i>	String. Optional. The name of the field where the cursor is moved.

If the displayed page contains no entry fields, leave the optional *field_name* empty (" ").

The keyboard is unlocked, permitting the operator to enter data.

If you forget to call SE_StartEntry after a SE_FirstPageRequired or SE_KeyPress events the terminal keyboard remains locked.

Example

In this example inside the SEFirstPageRequired event a call to SE_Page tells Sierra Engine to display the page "Login" as the first page every time a terminal is connected

Here oSE_Main is the instance of cSE_Main used by the application.

The example uses SE_FieldData to fill the field "Radio#.f" with the physical number of the terminal (also known as radio ID). The cursor is positioned on the entry field "Username.e" and the keyboard is unlocked using SE_StartEntry.

```
Private Sub oSE_Main_SEFirstPageRequired(terminal_Number As Byte)
    oSE_Main.SE_Page(terminal_Number) = "Login"
    oSE_Main.SE_ClearAllEntryFields terminal_Number
    oSE_Main.SE_FieldData(terminal_Number, "Radio#.f") = _
        oSE_Main.SE_TerminalID(terminal_Number)
    oSE_Main.SE_StartEntry terminal_Number, "Username.e"
End Sub
```

SE_TerminalID Property

Description Read-only. Get the terminal physical number.

Syntax [*SE_object_*]SE_TerminalID(*terminal_number*)

Remarks The SE_TerminalID property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

Returns integer between 0 and 3840. For *terminal_number* = 0 (virtual terminal) the SE_TerminalID always returns 0.

The physical number is the value assigned to *Cellular* to every terminal in the radio parameters. This number must be set in setup mode using the Terminal Status dialog.

SE_TSEFilename Property

Description Get or set the path & name of the screen definition file (.tse)

Syntax [*SE_object*]SE_TSEFilename(*terminal_number*) [=string]

Remarks The SE_TSEFilename property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

The SE_TSEFilename property must be set before calling to SE_StartEngine. Use the complete path to the file generated with the Teklogix's WinTSF utility.

Example

In this example an instance of cSE_Main is created with the name oSE_Main. The name and path of the .tse file is set using the property SE_TSEFilename and the Sierra Engine is started.

```
Set oSE_Main = New cSE_Main
oSE_Main.SE_TSEFilename = App.Path & "\myTSEfile.tse"
oSE_Main.SE_StartEngine
```

SE_UserLogged Property

Description Get or set the terminal user name.

Syntax [*SE_object_*]SE_UserLogged(*terminal_number*) [=string]

Remarks The SE_UserLogged property has the following parts:

Part	Description
<i>SE_object</i>	Instance of cSE_Main
<i>terminal_number</i>	Byte. The logical number of the terminal.

The user name passed to this property appears in the Terminal Status Dialog. The status of the terminal is changed to “Logged” and this change is registered in the log, further entries in the log referencing this terminal will use the user name supplied.

Appendix A: Enums

Alignment_Enum

ALIGN_CENTER = 0
ALIGN_RIGHT = 1
ALIGN_LEFT = -1

Icon_Enum

ICON_SYSTEM = 1
ICON_WARNING = 2
ICON_RED_TERM = 3
ICON_YELLOW_TERM = 4
ICON_GREEN_TERM = 5
ICON_RED_MAN = 6
ICON_YELLOW_MAN = 7
ICON_GREEN_MAN = 8
ICON_BLACK_MAN = 9

Mode_Enum

MODE_NORMAL = 1
MODE_CONFIG = -1

Key_Enum

KEY_HEY_YOU = -1	KEY_F11 = 11	KEY_F23 = 23
KEY_ENTER = 0	KEY_F12 = 12	KEY_F24 = 24
KEY_F1 = 1	KEY_F13 = 13	KEY_F25 = 25
KEY_F2 = 2	KEY_F14 = 14	KEY_F26 = 26
KEY_F3 = 3	KEY_F15 = 15	KEY_F27 = 27
KEY_F4 = 4	KEY_F16 = 16	KEY_F28 = 28
KEY_F5 = 5	KEY_F17 = 17	KEY_F29 = 29
KEY_F6 = 6	KEY_F18 = 18	KEY_F30 = 30
KEY_F7 = 7	KEY_F19 = 19	KEY_F31 = 31
KEY_F8 = 8	KEY_F20 = 20	KEY_F32 = 32
KEY_F9 = 9	KEY_F21 = 21	KEY_F33 = 33
KEY_F10 = 10	KEY_F22 = 22	KEY_F34 = 34

