Create Screen Saver in Limnor

Limnor ships with a Screen Saver performer. You may use it in your applications to add screen saver features.
The Screen Saver performer allows you to run different screen saver pages at different time. If you use it in a public kiosk, you can sell screen saver time to different advertisers.

Add Screen Saver performer

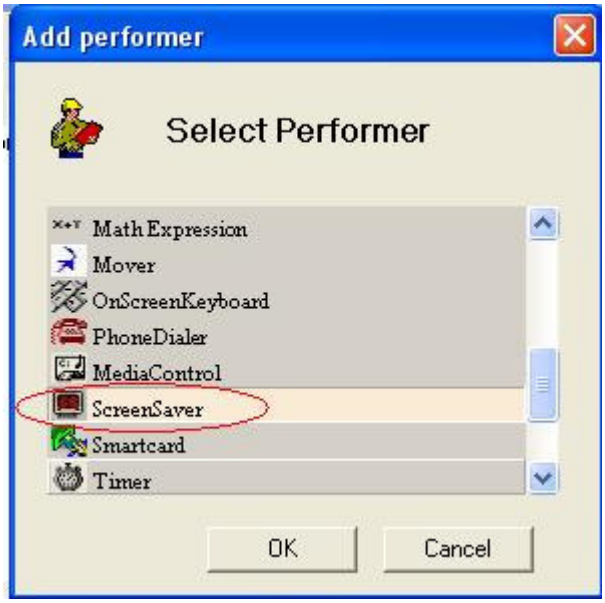
To add screen saver features to your Limnor application, add the Screen Saver performer to the Application performer.
To access the Application performer, click Project button or select Project Explorer menu

Project button: 

Project Explorer menu: 

The “Project browser” window appears. The Application performer is listed as the first item. To add the Screen Saver performer, select the Application performer and click “New: button:

It will ask you to select a performer type, select the Screen Saver performer, click OK button:



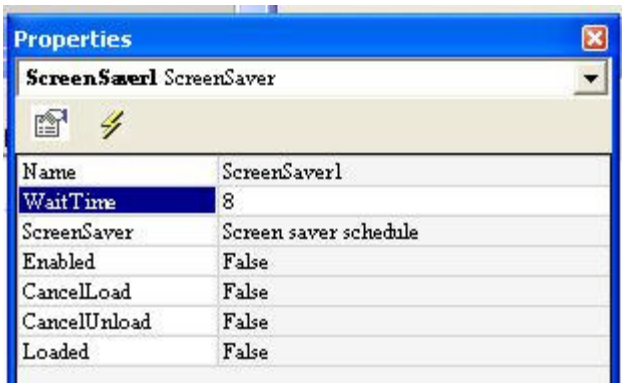
A new Screen Saver performer appears under the Application performer:

We need to set the properties of the screen saver performer to fit our needs.

Set Screen Saver properties

Right-click on the Screen Saver performer, select Properties:

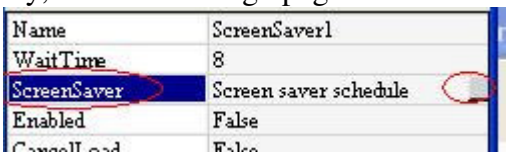
The properties of the Screen Saver appear:



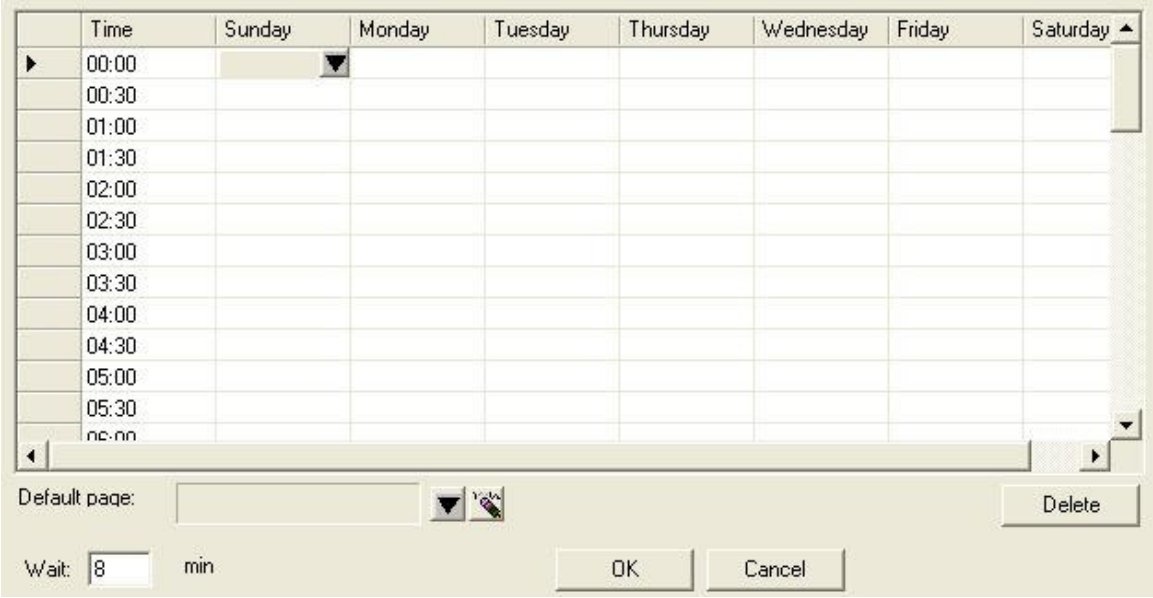
WaitTime – this property indicates how many minutes it will wait for keyboard and mouse activities before a screen saver page appears.
ScreenSaver – this property indicates all the screen saver pages. It allows you to set a different screen saver page every half hour in a week. So you have 7 x 48 = 336 time slots to sell advertisements. We will talk about it in more details in the next section.
Enabled – this property indicates if the screen saver is enabled or not. If this property is set to True, the Screen Saver performer will start watch for mouse and keyboard activities. If there is not a mouse or keyboard activity for a certain time, indicated by WaitTime property, a screen save page is displayed. If this property is set to False, the Screen Saver will not watch for mouse and keyboard activities.
CancelLoad – This is a write-only property used at the BeforeScreenSaverOn event. When the Screen Saver performer does not detect mouse and keyboard activities in a certain time, it will fire the BeforeScreenSaverOn event. If you use an action to set CancelLoad property to True in responding to BeforeScreenSaverOn event, the Screen Saver will not display the screen saver page and start watching for mouse and keyboard activities again.
CancelUnload – This is a write-only property used at the BeforeScreenSaverOff event. When a screen saver page is displayed and the Screen Saver performer detects a mouse and keyboard activity, it is suppose to close the screen saver page. Before closing the screen saver page, it will fire the BeforeScreenSaverOff event. If you use an action to set CancelUnload property to True in responding to BeforeScreenSaverOff event, the Screen Saver will not close the screen saver page, and keep watching for another mouse or keyboard activity.
Loaded – This is a read-only property. If it is true, then a screen saver page is currently on display. If it is False, no screen saver page is on display.

Assign Screen Saver pages

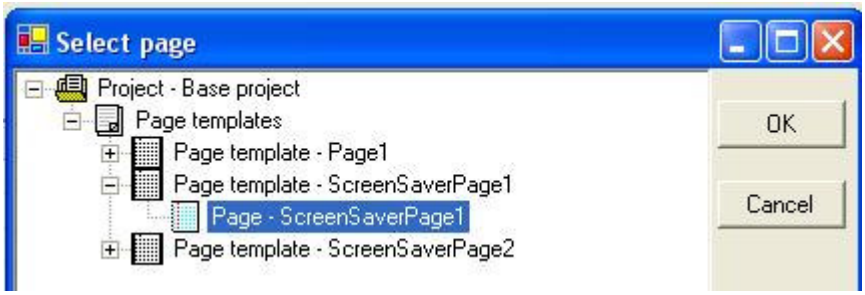
Select ScreenSaver property, click  to assign pages as screen saver pages:



It will show every half hour in a week as a time slot for a screen saver page. Each time slot can be a unit to be sold to one of your advertisers.



Select each time slot, click ▼ button to select a page as the screen saver page for the time slot:

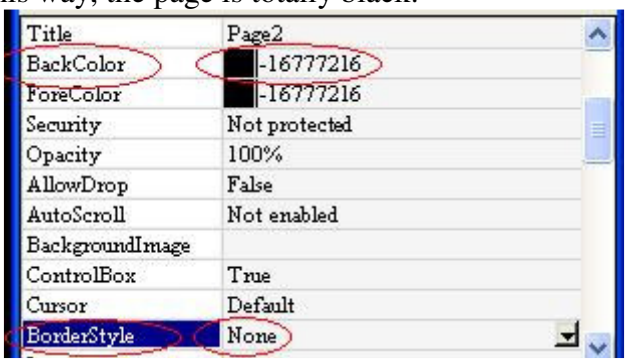


You may also specify a “Default page”. When a time slot is not assigned a screen saver page, the “Default page” will be used as its screen saver page.
If you want to remove a screen saver page from a time slot, select the time slot, click “Delete” button.

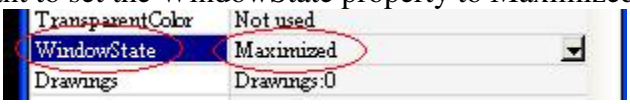
Create Screen Saver pages

We have been talking about screen saver pages above. What is a screen saver page? You may use any page in your Limnor application as a screen saver page.
But you have to remember that a screen saver page should not be a static page if it is not totally black, because a static page will not save your screen.
The Screen Saver performer does not try to detect and warn you if you use a static page. Therefore if you want to use a page as screen saver page, it is your responsibility to make sure it is not static.
How to make a page not static? That is, there is not a single point which is not black and never changes. There are many ways. For example, you may let the page plays a video continuously; use a timer to repeatedly show a set of pictures one by one; use a timer to repeatedly show a set of pages one by one; etc.

In this sample, we use a Mover performer to move a Label performer randomly on a page with black background.
As a rule, we should always set the BackColor property to Black and BorderStyle property to None. This way, the page is totally black:

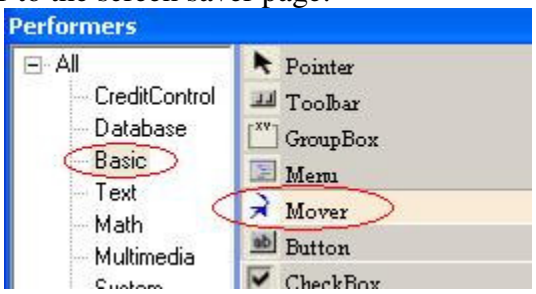


Usually you may want to set the WindowState property to Maximized:

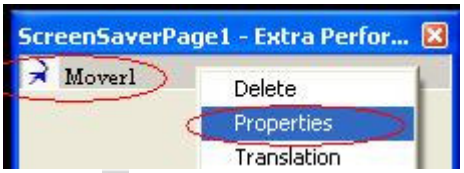


Now drop a Label performer on the screen saver page and set its properties to fit our needs:

Set BackColor property to Black, set BorderStyle to None. Set Text property to a text we want to show, and set Font to the one we like.
We will use a Mover performer to move this label randomly on the page.
Drop a Mover performer to the screen saver page:

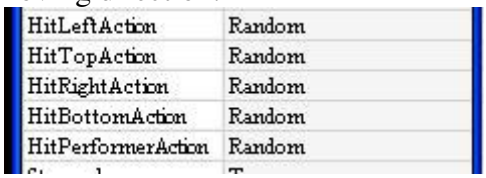


A Mover performer appears in the “Extra performer” window. Right-click the Mover performer, select “Properties”:



Select “Performer” property, click , then select the Label performer:

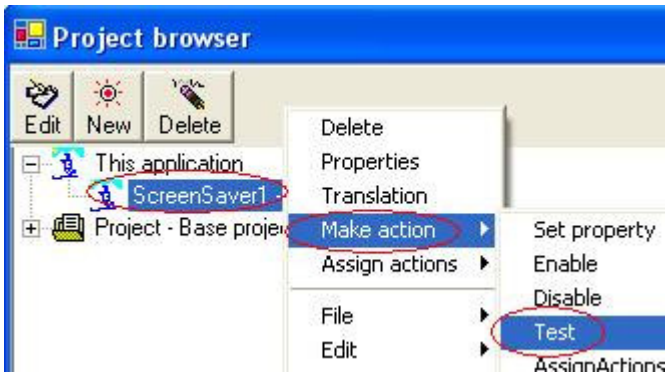
This way, we tell the Mover performer to move the label.
We set the "HitLeftAction" property to Random. That means, when the label hits the left edge of the page, it will randomly change its moving direction. Actually, we set all "Hit*Action" properties to Random. That is, when the label hits any side of the page, it will randomly change its moving direction:



We set "AutoStart" property to True so that when the page is loaded, the moving starts immediately.
Now we need to set "Speed" and "RefreshRate" properties. "Speed" property indicates how many pixels per second the performer will be moved. "RefreshRate" property indicates how many milliseconds the performer be redrawn on the screen. The bigger the "Speed" property is, the faster the performer is moving. The smaller the "RefreshRate" is, the smoother the moving will appears. You have to experiment with various values for these two properties to see it moves in a way you like. You may use the Test method of the Screen Saver performer to experiment on your settings, see the next section

Test Screen Saver pages

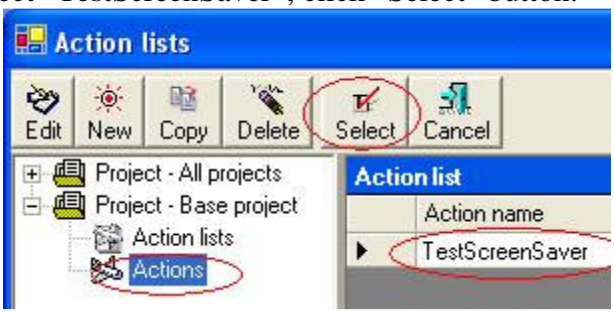
We have finished designing the screen saver pages. We may run the program, do not touch mouse and keyboard and wait for timeout and the screen saver page will appear. We can see if the screen saver appear as we wanted, for example, the moving of performers is at right speed. But if we test our screen savers this way, that will be time consuming because we have to wait for timeout.
The Screen Saver performer has a Test method, which may be used to force the application enter screen saver mode and open the screen saver page.
Create a test action – Open the Project Browser window; select the Screen Saver performer under Application; right-click on the Screen Saver performer; choose "Make action"; choose "Test":



Give an action name, say, TestScreenSaver:

Assign TestScreenSaver action -- Now create a button on the home page, right-click on the button; choose “Assign actions”; choose “Click”:

Select “Actions”; select “TestScreenSaver”; click “Select” button:



We are done. Press F2 to run the program and click the button to test the screen saver. Press F2 again to go back to design mode and continue our development.

=== EOF ===