1 Introduction

One of the surprise new features in Xara Xtreme Pro is the ability to generate basic Flash animations. Flash is vector based animation and it is similar to creating animated gifs. Xara's implementation of Flash is slick, intuitive, and unique. In Xara, the process is to design the "key" frames (a key frame is like a snapshot in time of the animation sequence). Flash then generates the in-between stages of animation.

So for example, in your first key frame you may draw a simple object, and you then copy that frame to generate a frame 2. In that second key frame, you may, say, move the object to a new position. The exported animation consisting of two drawn frames will then fully animate the shape moving from its start position in key frame 1 to the end position in key frame 2 without having to manually draw each in-between frame like you would in an animated gif. It is typically much faster to create animations in this way.

Flash however is very limited in what it can do when it comes to generating these intermediate steps. It is by no means as versatile or advanced as Xara's blend tool, nor does it support most of Xara's advanced features, such as vector feathering. There are strict criteria that must be followed for successful "tweening" in the animation (tweening is the process when Flash automatically generates the in-beTWEEN steps between key frames). You are allowed to do the following for successful tweening between two key frames:

1. **Move**: you can change the object position.
2. **Scale**: you can enlarge or reduce objects in size.
3. **Squash or skew**: you can squash or skew objects.
4. **Rotate**: you can rotate objects.
5. **Transparency**: you can adjust the overall (flat) transparency of the object.
6. **Colour transform**: you can change the fill (flat) colour of objects.

This is all Flash currently supports, and if you try to do anything else, you'll get errors and your animation won't be generated. As you can see, Flash is limited in what it can do, but what it lacks in animation features is made up for in low file size, higher animation frame rates, and reduced time to create the animation.

Videos associated with this tutorial may possibly be found in future on YouTube under user XaraXhris. Search for Xara clips. Some images in this document can also be double clicked and opened in Xara; this is a new feature of Xara Xtreme Pro.

2 Creating a Flash animation

The best way to learn Flash is to watch someone else use it. See video "**Creating a Flash animation**".

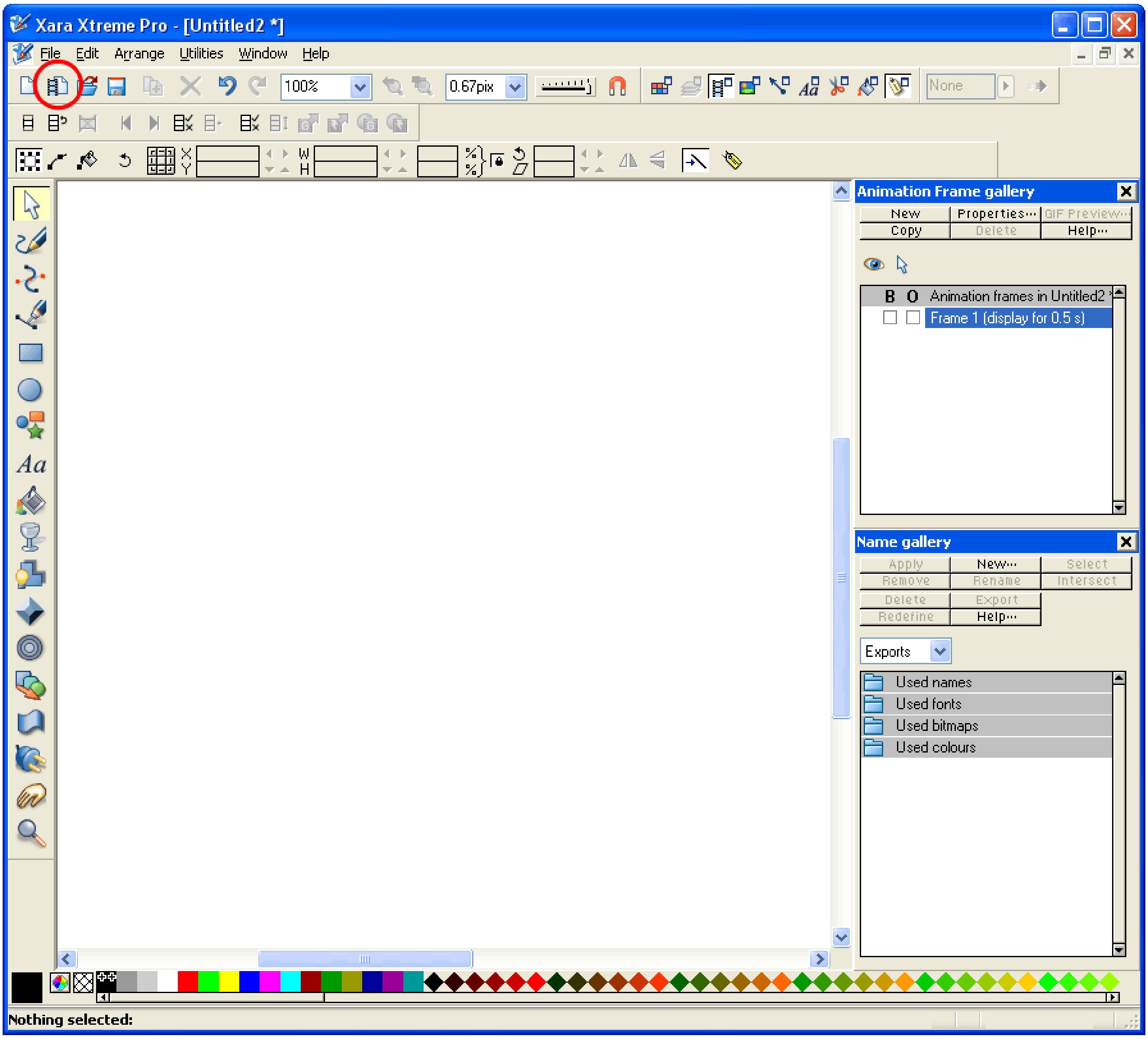
To quote from Xara's help file, the procedure for creating Flash animations is as follows:

1. Draw your first frame and name any objects that you want to animate.
2. Create a new key frame, usually done by copying the earlier frame.
3. Move and transform the objects as required.
4. Preview the results.
5. Repeat steps 2 through 4.

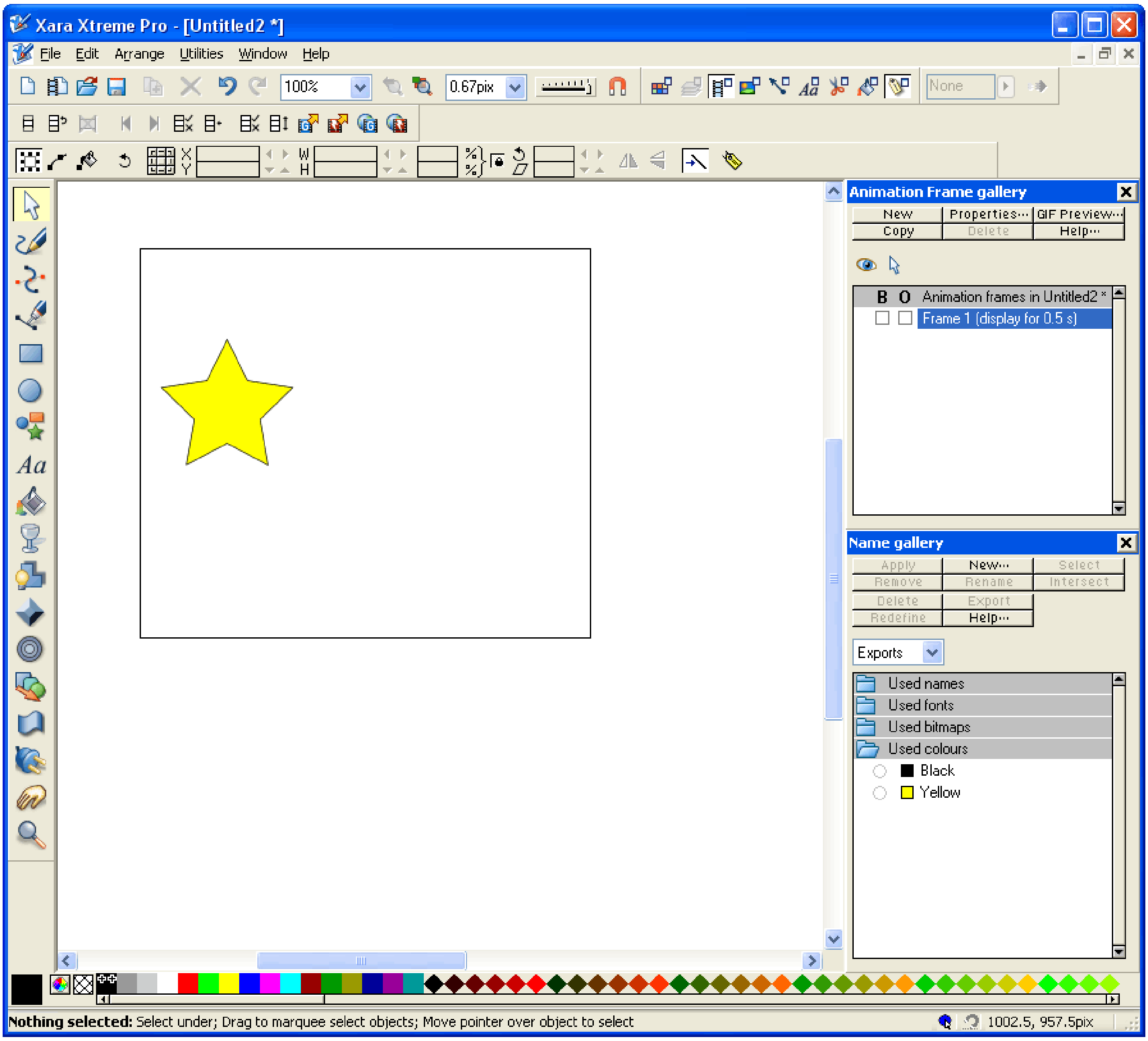
Making Flash animations in Xara teaches you well how to use the name gallery – one of Xara's less discussed, but powerful features. More on this is shown in Section 5.3.

So let's now generate a simple Flash animation which highlights basic techniques. It's recommended that you have both the frame and name galleries open while creating Flash animations to make things easier. They can be docked at the right-hand side out of the way.

1. Click on the new animation button on the toolbar to start a new animation document.

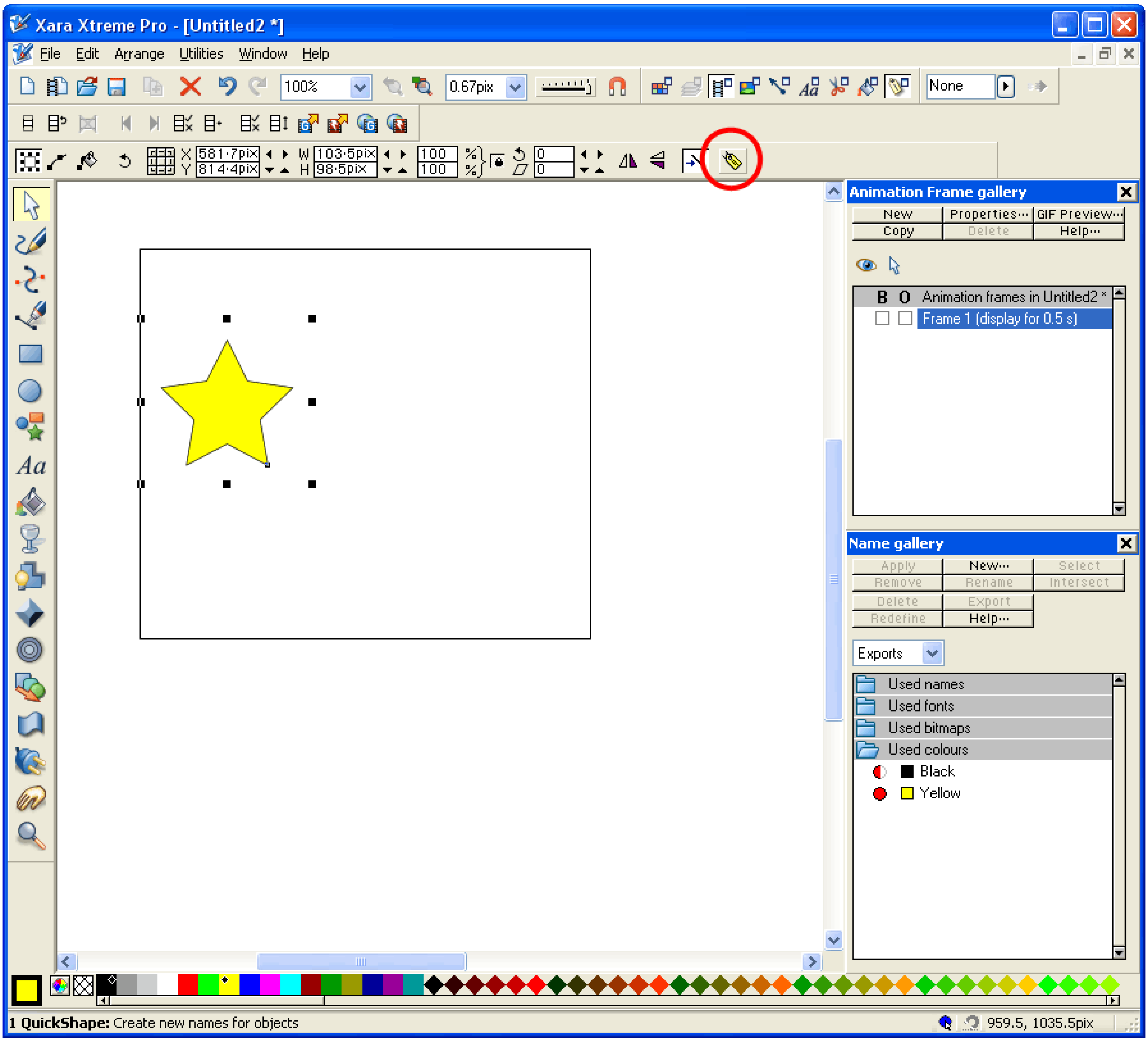


2. It is recommended that animations be drawn on top of a larger object that can be used as a board or frame to more easily allow you to determine the physical size of your animation. Create this frame and a basic shape as shown below.



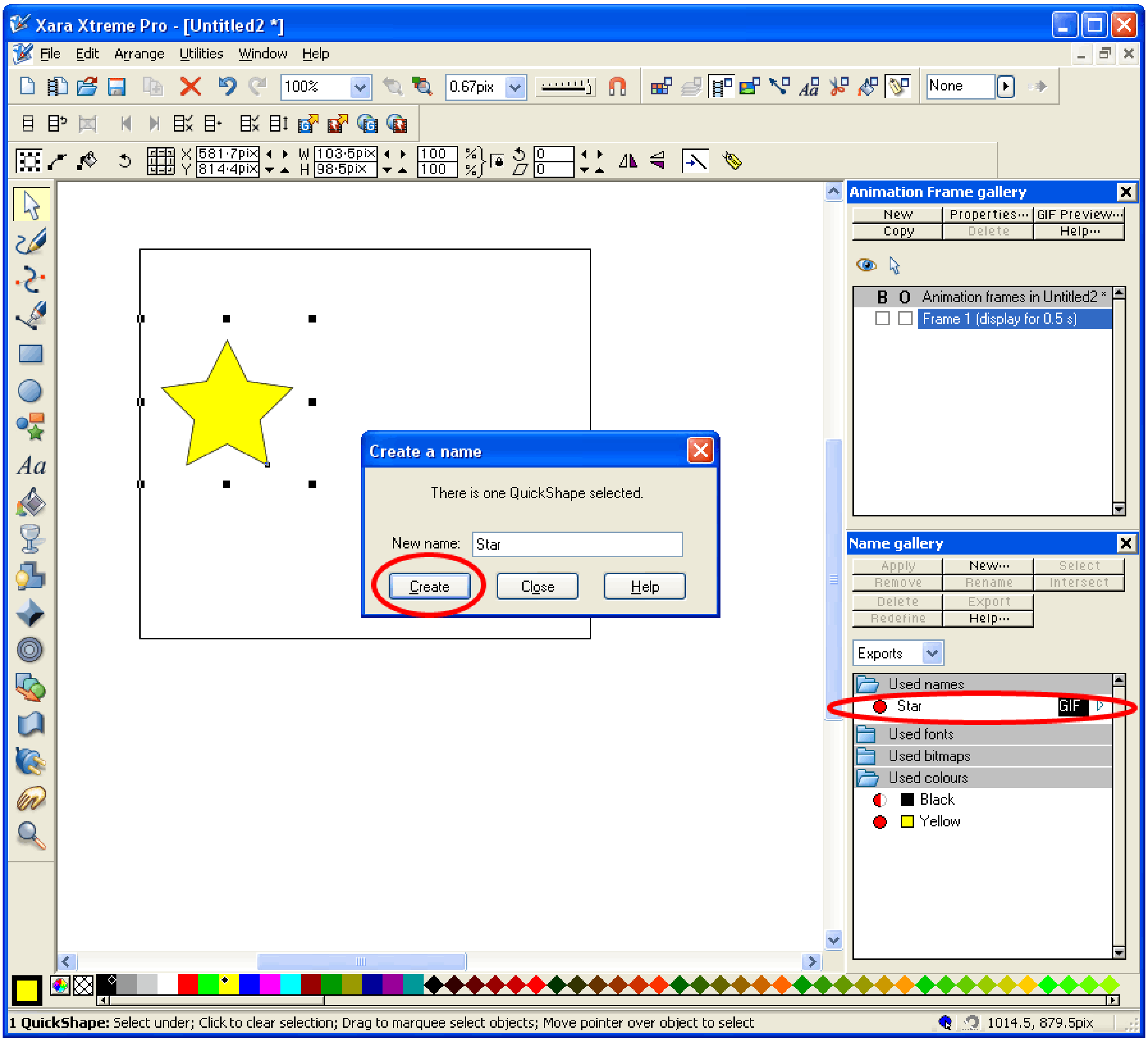
Nothing selected: Select under; Drag to marquee select objects; Move pointer over object to select

3. Select the star to be animated and then click the “create new names” button on the selector tool infobar. Note that we don’t want the border to animate and so it’s not given a name.



1 QuickShape: Create new names for objects

4. Enter an appropriate name in the dialogue box that appears, such as “Star”, and click ok. The name now appears in the name gallery as shown below, and it is assigned to the object. The red circle denotes that the selected object(s) (our star in this case) has that name, and the full circle (instead of half-circle or empty circle) indicates that the object is the only object in the document to have that name. This completes the first key frame.



Create a name

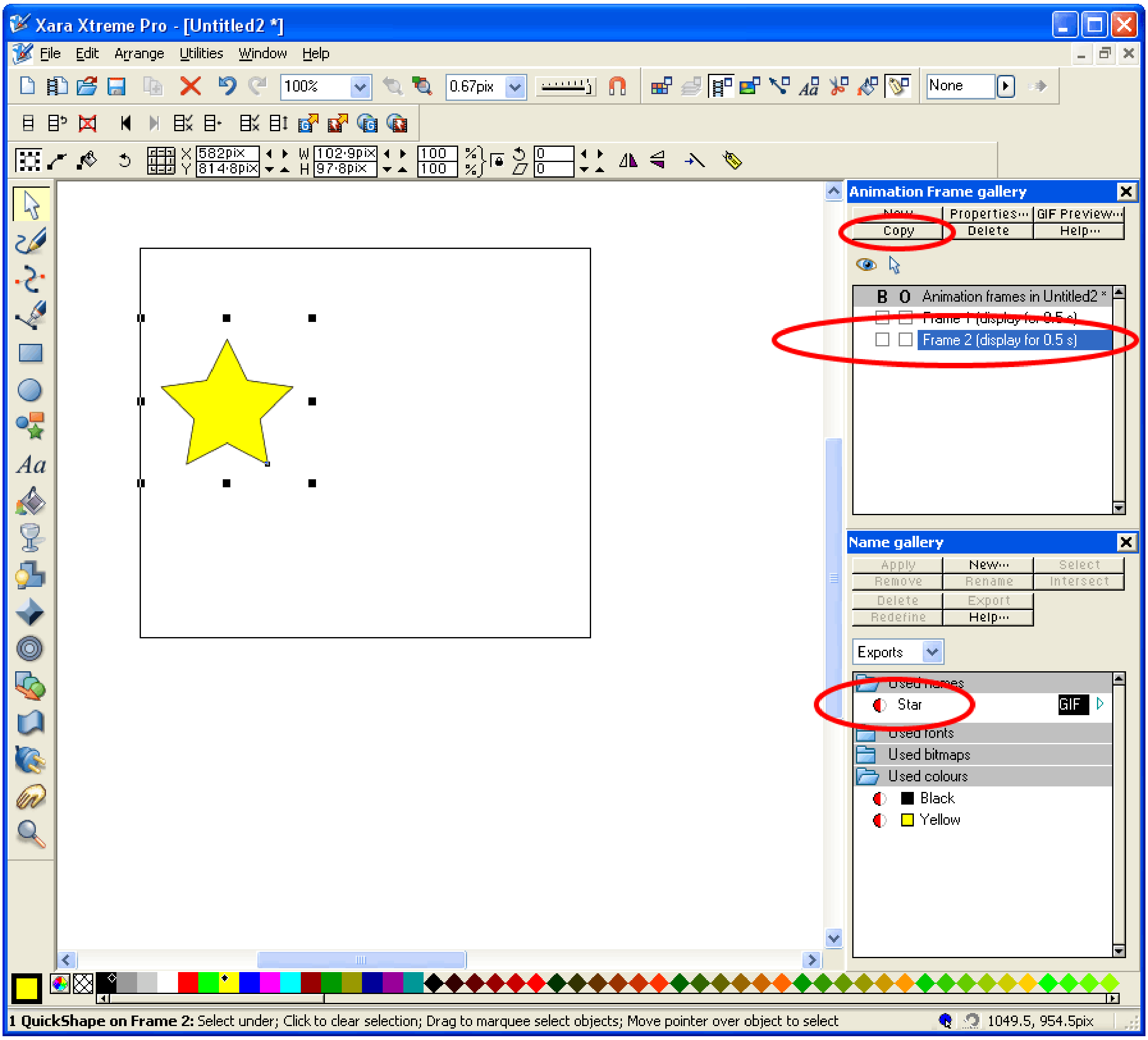
There is one QuickShape selected.

New name: Star

1 QuickShape: Select under; Click to clear selection; Drag to marquee select objects; Move pointer over object to select

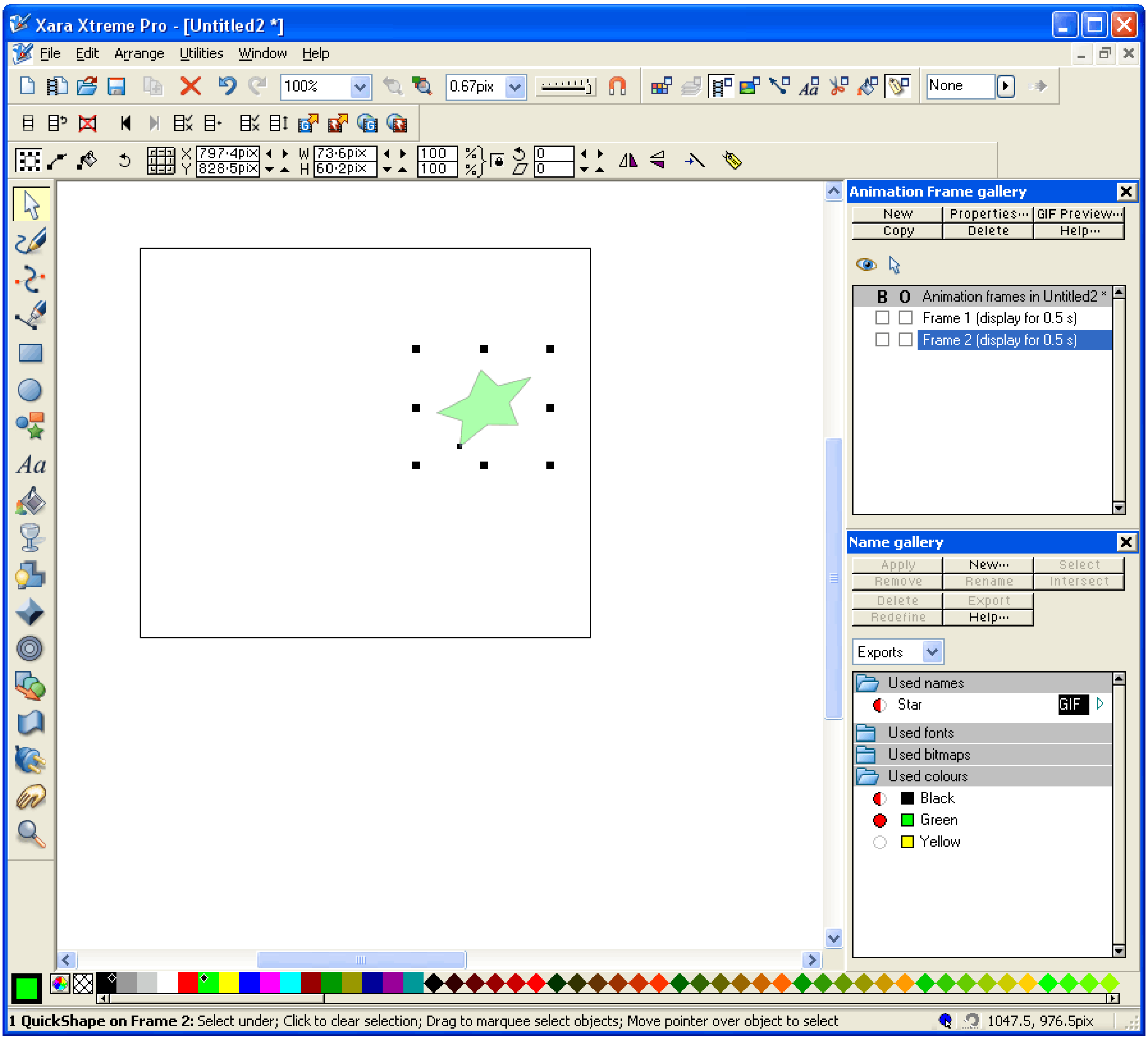
1014.5, 879.5pix

5. We now need to generate the next key frame. This example is just a simple two-frame animation, and so this frame should represent the final look of our object after we've animated it (making sure we follow all the Flash rules shown above). Click the copy button in the frame gallery as shown below. This copies Frame 1 and makes a new frame called Frame 2. Notice how the red circle next to the name "Star" is now a half-circle. This is because our yellow star now exists twice in the document, and each time (on each frame) has the name "Star".



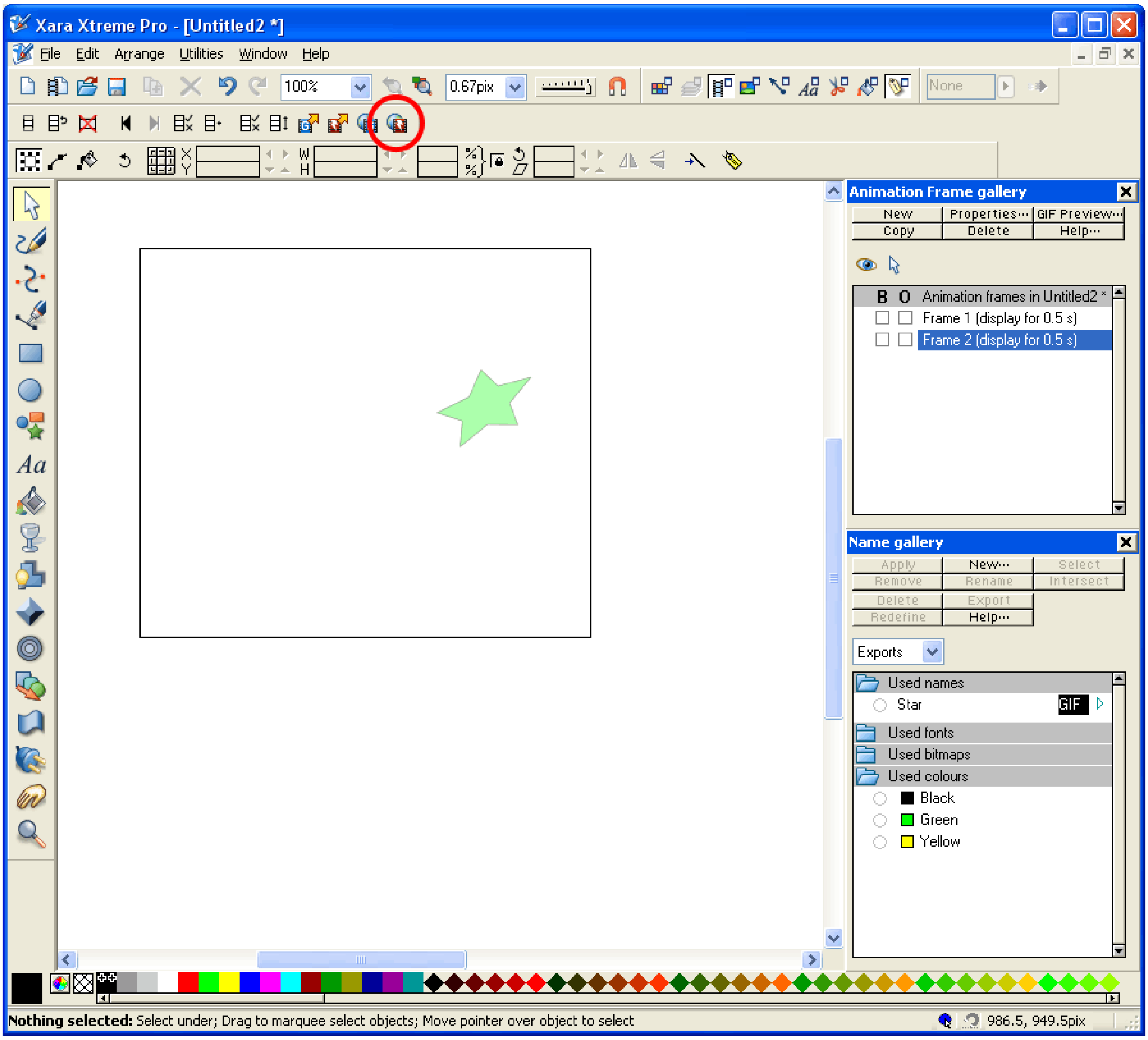
1 QuickShape on Frame 2: Select under; Click to clear selection; Drag to marquee select objects; Move pointer over object to select 1049.5, 954.5pix

6. Apply the desired (allowable) transformations to the star that you want to be animated. For example, we will perform all 6 possible transformations listed above: move, scale, squash/skew, rotate, change colour, change (flat) transparency value, as shown below.



1 QuickShape on Frame 2: Select under; Click to clear selection; Drag to marquee select objects; Move pointer over object to select 1047.5, 976.5pix

7. The animation is now complete. To preview the animation, press the "preview Flash animation in browser" button on the animation toolbar as shown below.



B O Animation frames in Untitled2 *

□ □ Frame 1 (display for 0.5 s)

□ □ Frame 2 (display for 0.5 s)

Exports

Used names

○ Star GIF

Used fonts

Used bitmaps

Used colours

○ ■ Black

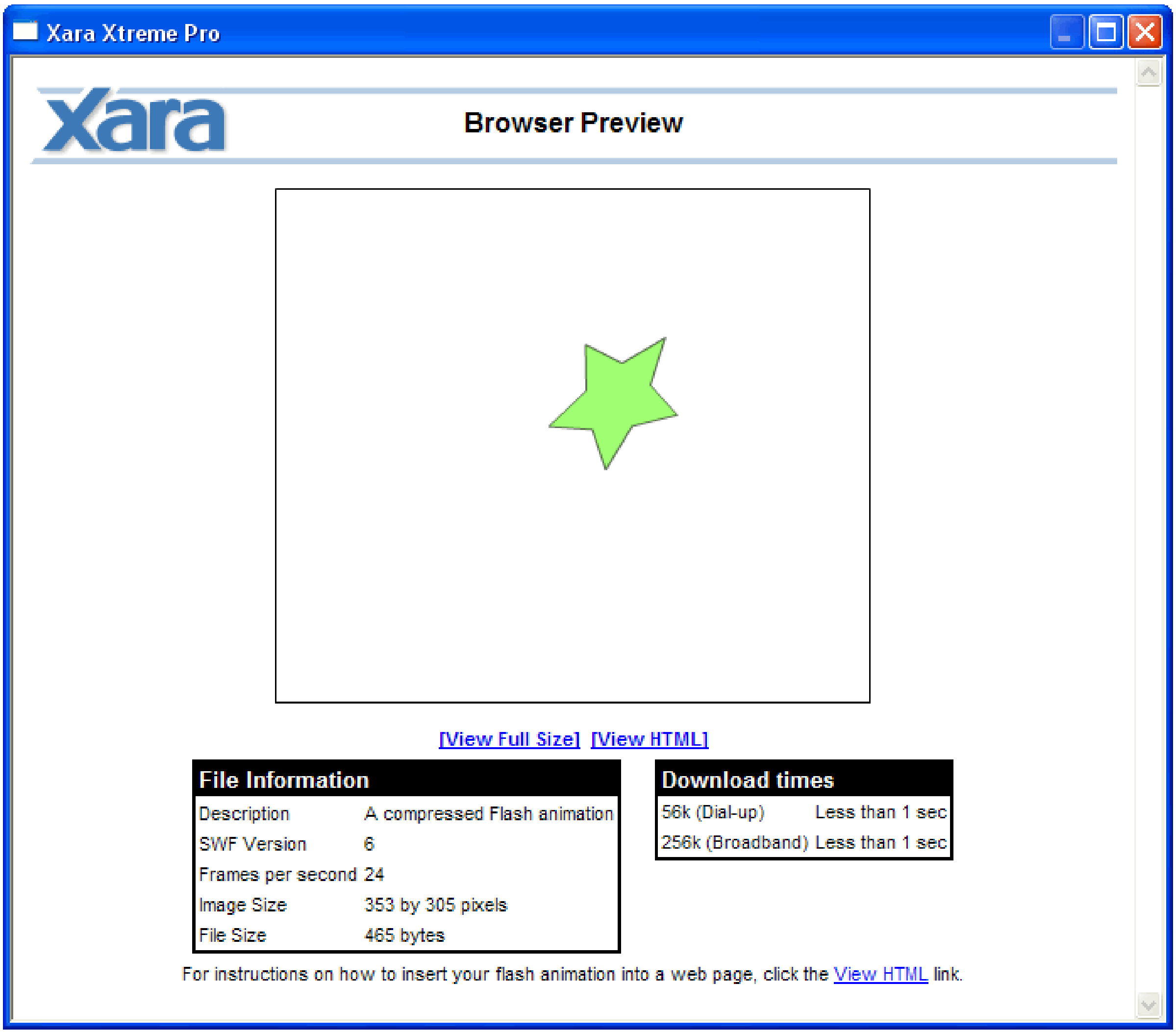
○ 🟩 Green

○ 🟨 Yellow

Nothing selected: Select under; Drag to marquee select objects; Move pointer over object to select

986.5, 949.5pix

The screen below appears showing you the animation and some important details about it, such as file size (notice how small it is) and exported Flash version etc.



Browser Preview

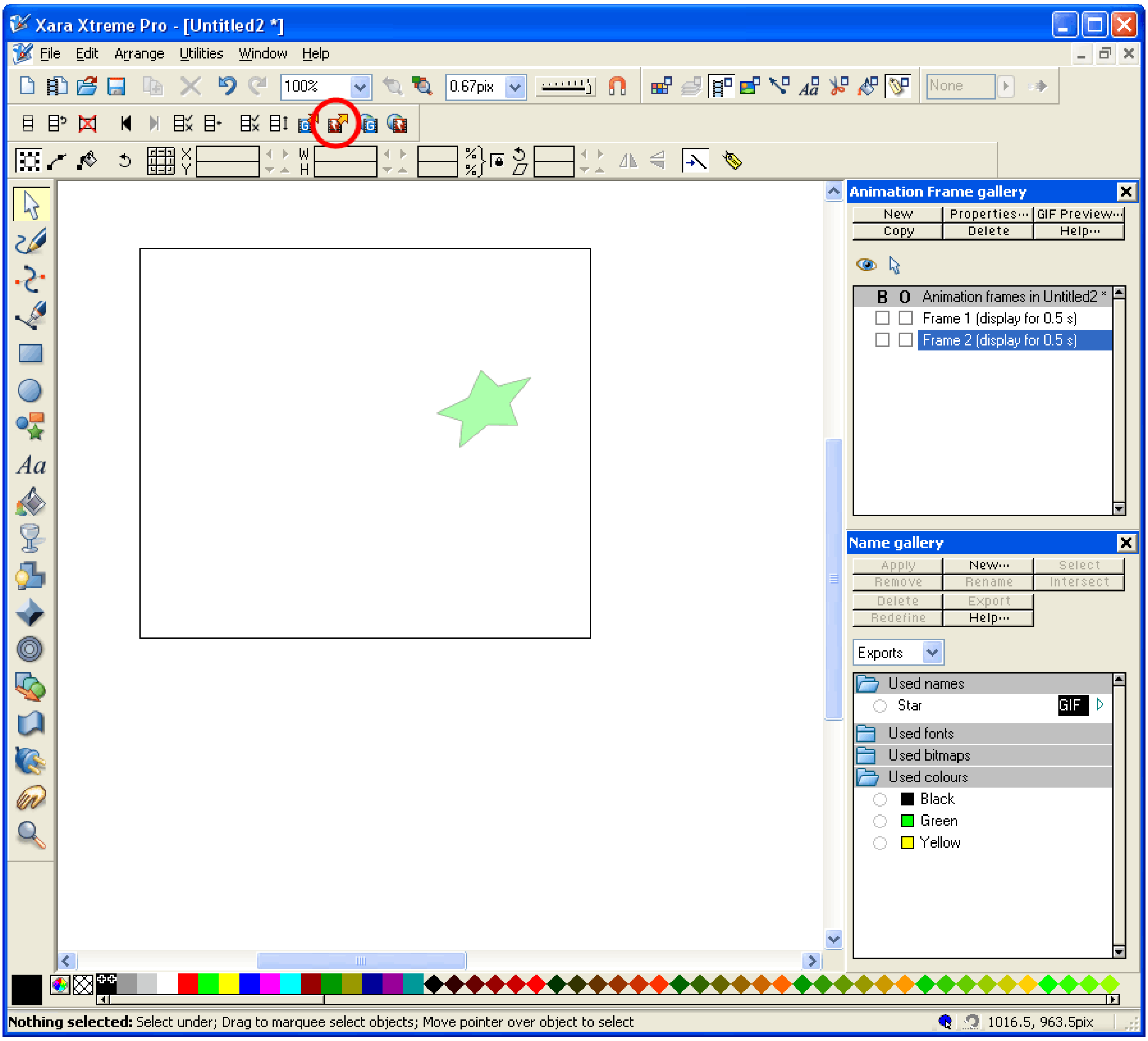
[View Full Size] [View HTML]

File Information	
Description	A compressed Flash animation
SWF Version	6
Frames per second	24
Image Size	353 by 305 pixels
File Size	465 bytes

Download times	
56k (Dial-up)	Less than 1 sec
256k (Broadband)	Less than 1 sec

For instructions on how to insert your flash animation into a web page, click the View HTML link.

Pressing the preview button will display any errors that may exist in more complex animations. The Flash animation can now be exported by selecting the export button on the same toolbar.



Nothing selected: Select under; Drag to marquee select objects; Move pointer over object to select

1016.5, 963.5pix

3 Commands

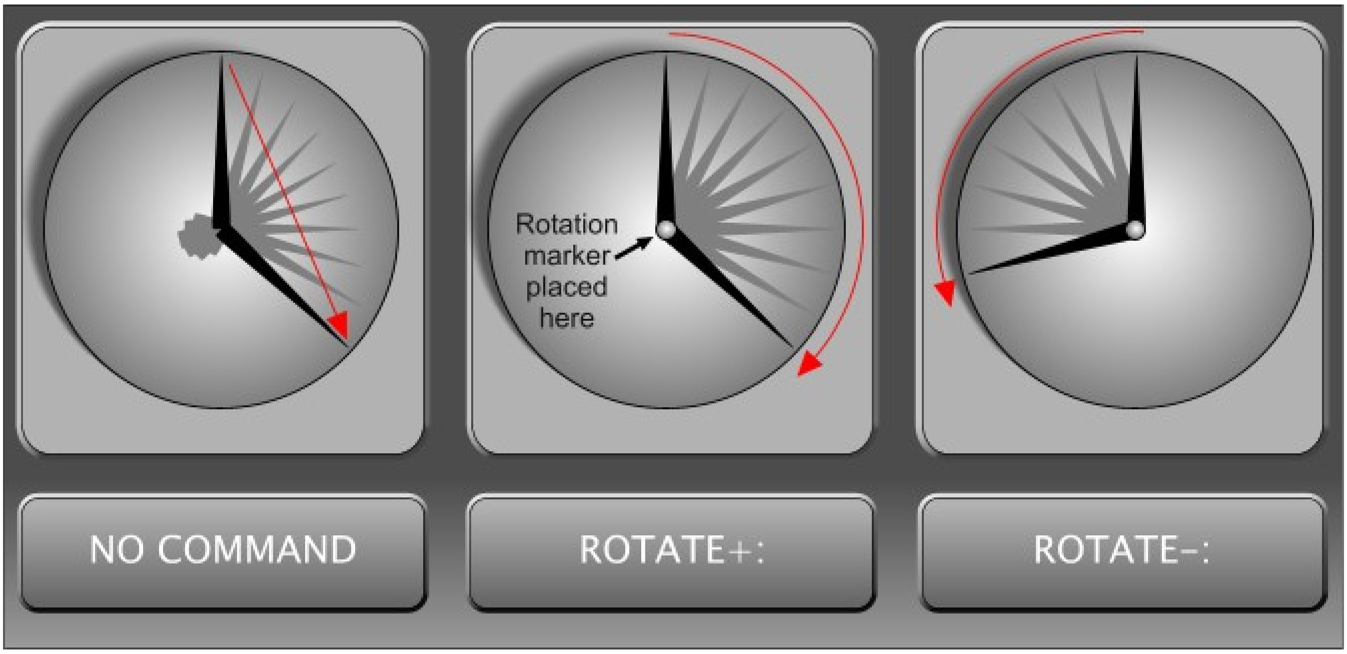
Commands can be used in Xara Flash to provide additional effects to animations. Commands can be applied to object names, the frames objects are on, or on the objects themselves. Each will be covered in their own section below.

3.1 Applied to names

3.1.1 Rotation:

(See video "**Name command – rotation**".) Objects rotate arbitrarily as they translate when tweened. However this isn't always desirable; the most obvious example is if you want to animate a clock hand – the hands should rotate about their ends near the centre of the clock instead of the centre of the arm.

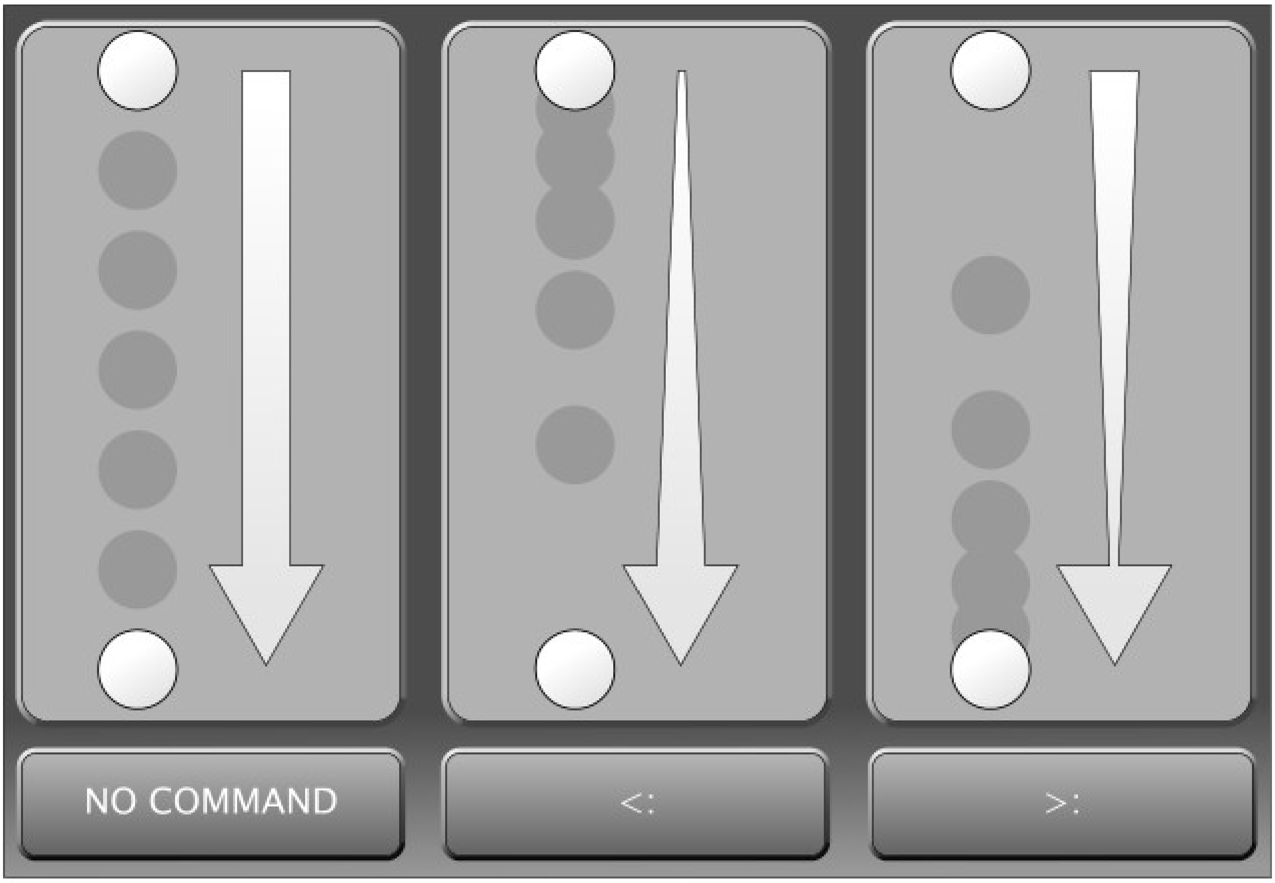
When naming an object, putting the text “rotate+:” or “rotate-:” before the object name will force the object to rotate about the position of the rotation marker, rather than transforming to the new position (slowly click twice on an object with the selector tool to view the rotation marker). “+” means the object will rotate clockwise, “-“ means anticlockwise as shown below. Colons denote the end of a command applied to an object name.



Tip: It’s possible to make an object rotate a full 360° in just one frame by using the rotate command and not changing the position of the object in a subsequent frame. However, the object will only rotate about its centre. This might have some uses in certain situations.

3.1.2 Acceleration:

(See video “**Name command – acceleration**”.) Objects can be made to accelerate when translating by putting “<:”, “>:”, “<>:”, or “><:” at the beginning of the object’s name in a similar fashion to the rotation command discussed above. Think of the chevron representing the object’s speed. For example, the shape “<” starts off small and gets larger – this is also how the object’s speed will change; it will start slow and become fast as shown below. Colons denote the end of a command applied to an object name. Note that the same object will start it’s acceleration over again at the start of each frame if it’s used in successive frames.



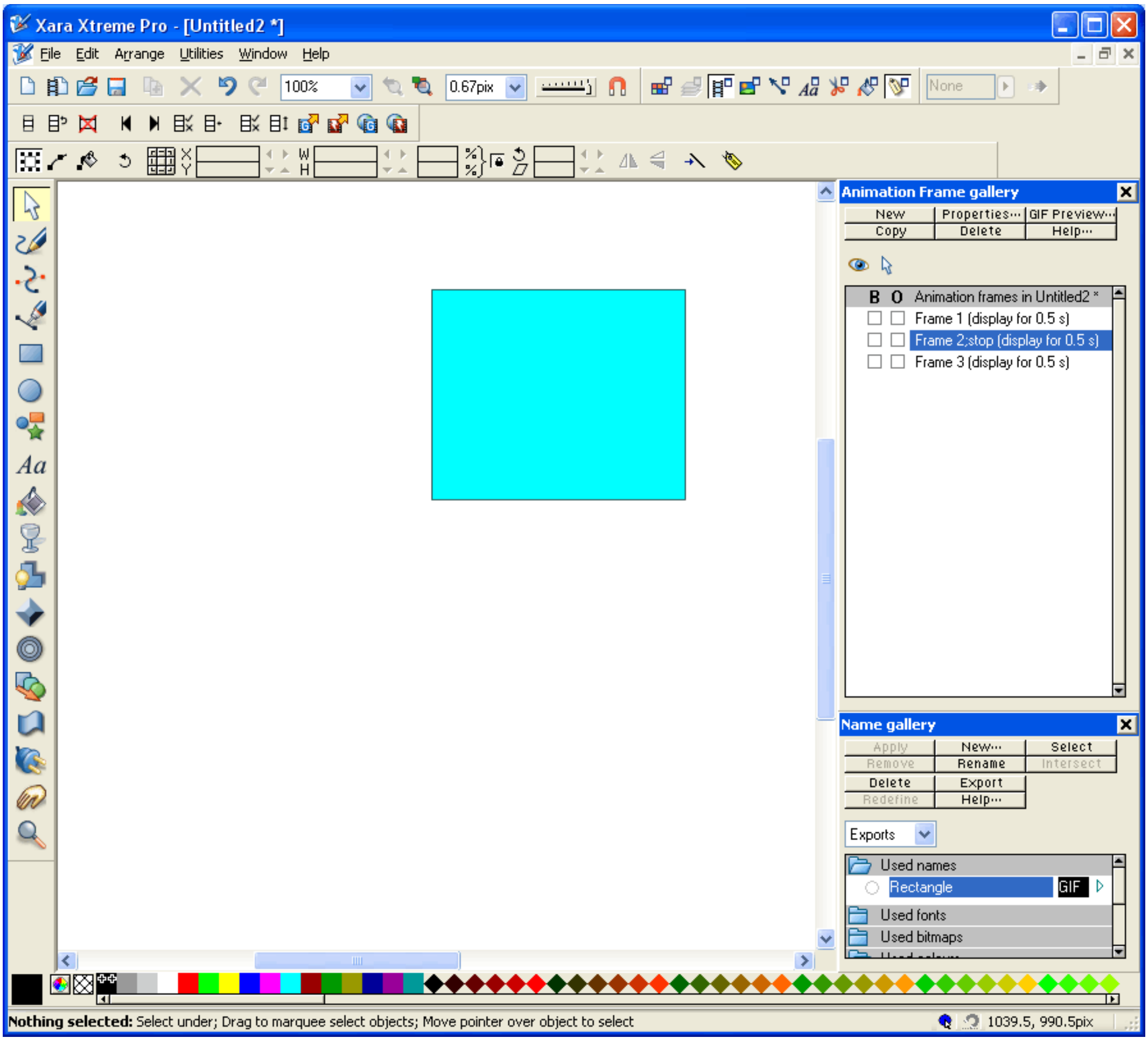
3.1.3 Summary:

Command	Description
rotate+:	Object rotates clockwise about rotation marker
rotate-:	Object rotates anticlockwise about rotation marker
<:	Object accelerates (starts slow, ends fast)
>:	Object decelerates (starts fast, ends slow)
<>:	Object accelerates then decelerates (starts slow, goes fast, ends slow)
><:	Object decelerates then accelerates (starts fast, goes slow, ends fast)

3.2 Applied to frames

3.2.1 ;stop

(See video "**Frame command – stop**".) Change the name of a frame to "<name>;stop" to have the Flash animation stop, once it has completed that frame. In the image below, the animation will run up to how things look in frame 2, but stop before progressing onto frame 3. This has use if you are planning on animating non-linearly through the frames (see goto command below).



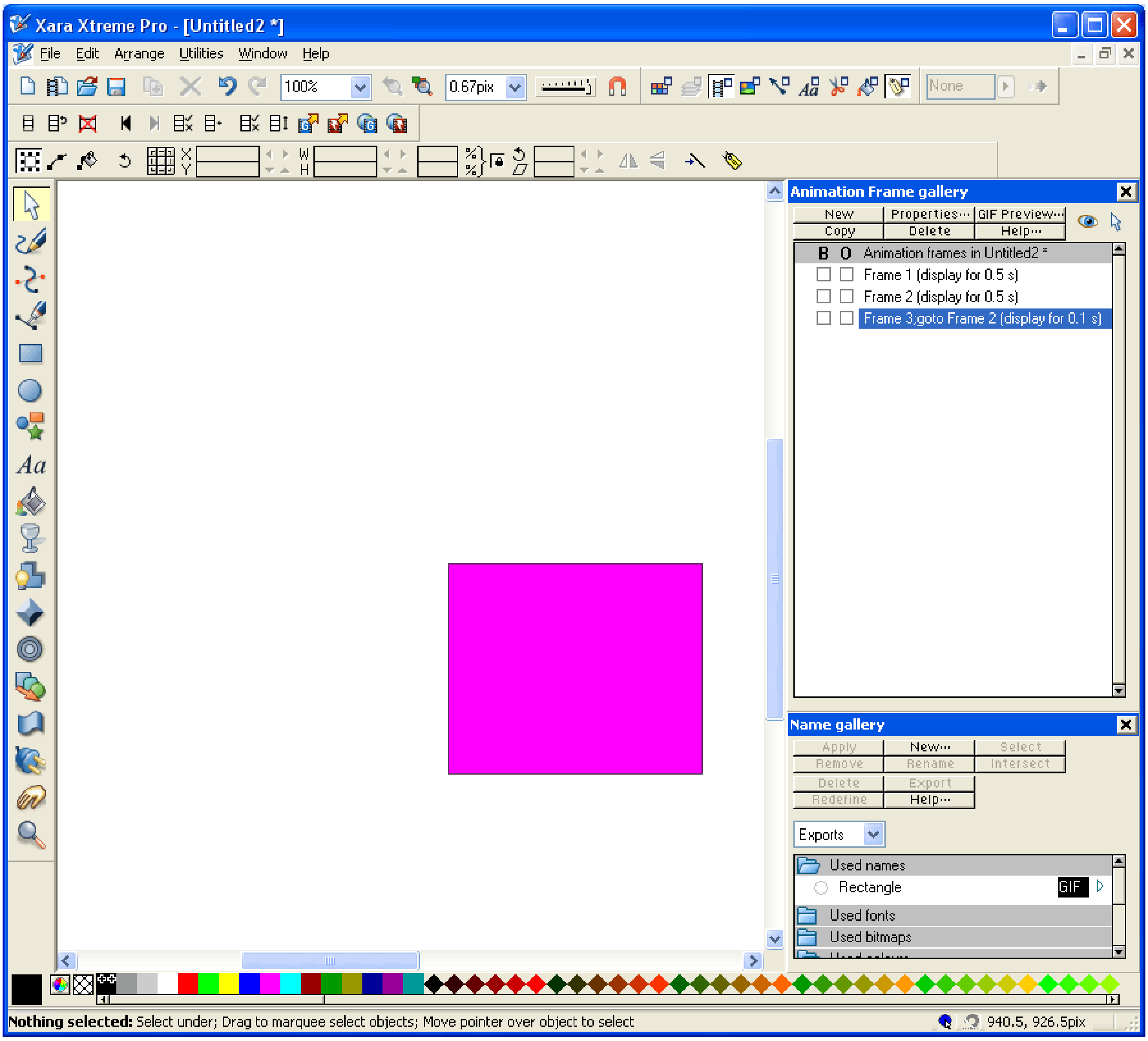
B O Animation frames in Untitled2 *
□ □ Frame 1 (display for 0.5 s)
□ □ Frame 2;stop (display for 0.5 s)
□ □ Frame 3 (display for 0.5 s)

Used names
○ Rectangle GIF
Used fonts
Used bitmaps

Nothing selected: Select under; Drag to marquee select objects; Move pointer over object to select 1039.5, 990.5pix

3.2.2 ;goto

(See video "**Frame command – goto**".) Change the name of a frame to "<name>;goto <desired frame name>" to have the Flash animation jump to the name of the frame specified once that frame has finished animating. In the image below, the animation will reach the final frame, then goto the previous frame before continuing – an endless loop of the last few frames.

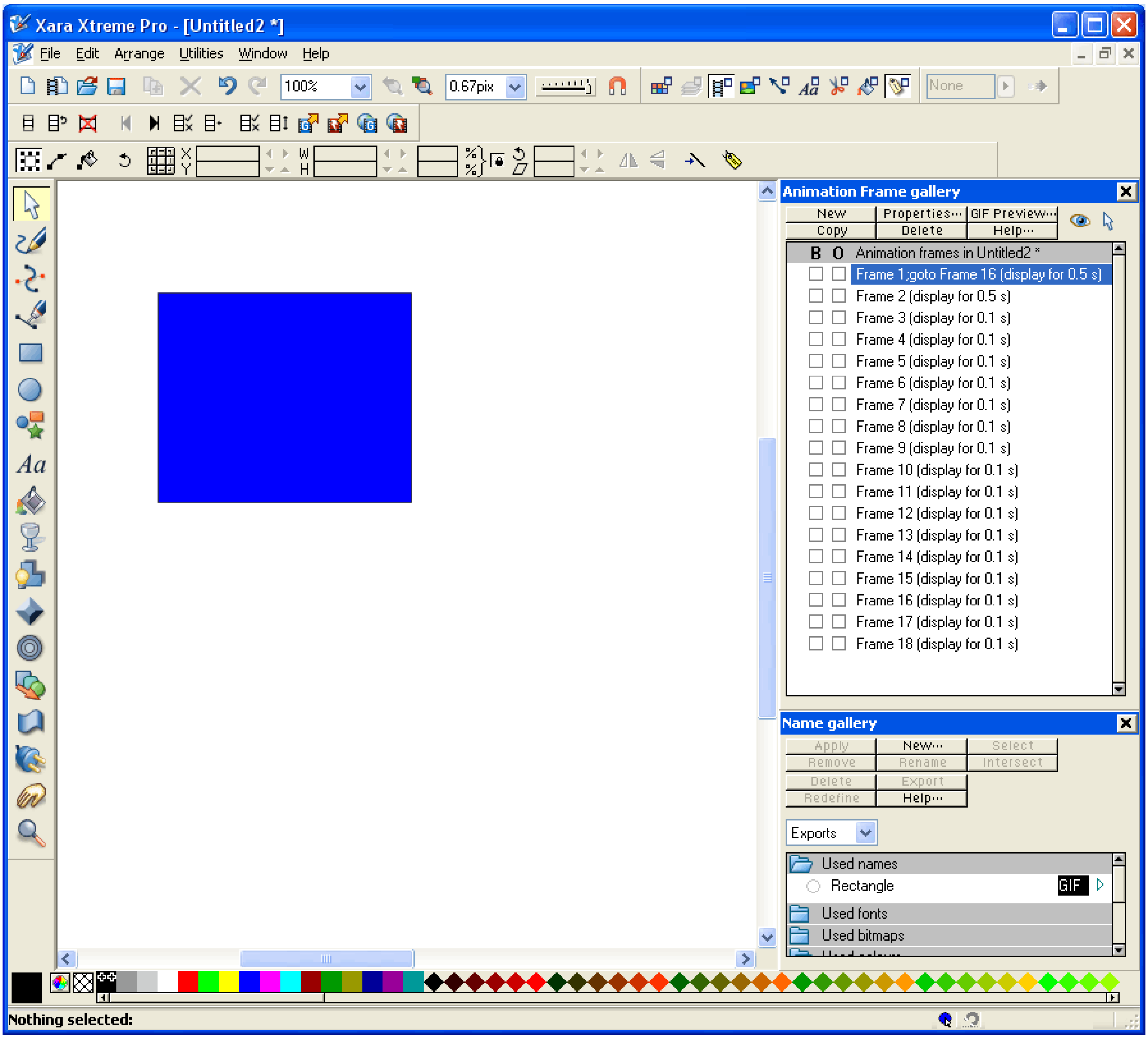


B	O	Animation frames in Untitled2 *
□	□	Frame 1 (display for 0.5 s)
□	□	Frame 2 (display for 0.5 s)
□	□	Frame 3;goto Frame 2 (display for 0.1 s)

Nothing selected: Select under; Drag to marquee select objects; Move pointer over object to select

940.5, 926.5pix

Tip: Use the goto command on the first frame to jump to the frame you want to preview when you are creating long animations. This helps you avoid watching an entire animation if you only want to see what an end section will look like for example. In the example below, the animation begins by jumping from frame 1 to frame 16 to watch the last few frames of animation instead of having to watch all frames.



B O Animation frames in Untitled2 *

□ □ Frame 1;goto Frame 16 (display for 0.5 s)
□ □ Frame 2 (display for 0.5 s)
□ □ Frame 3 (display for 0.1 s)
□ □ Frame 4 (display for 0.1 s)
□ □ Frame 5 (display for 0.1 s)
□ □ Frame 6 (display for 0.1 s)
□ □ Frame 7 (display for 0.1 s)
□ □ Frame 8 (display for 0.1 s)
□ □ Frame 9 (display for 0.1 s)
□ □ Frame 10 (display for 0.1 s)
□ □ Frame 11 (display for 0.1 s)
□ □ Frame 12 (display for 0.1 s)
□ □ Frame 13 (display for 0.1 s)
□ □ Frame 14 (display for 0.1 s)
□ □ Frame 15 (display for 0.1 s)
□ □ Frame 16 (display for 0.1 s)
□ □ Frame 17 (display for 0.1 s)
□ □ Frame 18 (display for 0.1 s)

Used names
○ Rectangle GIF
Used fonts
Used bitmaps

Nothing selected:

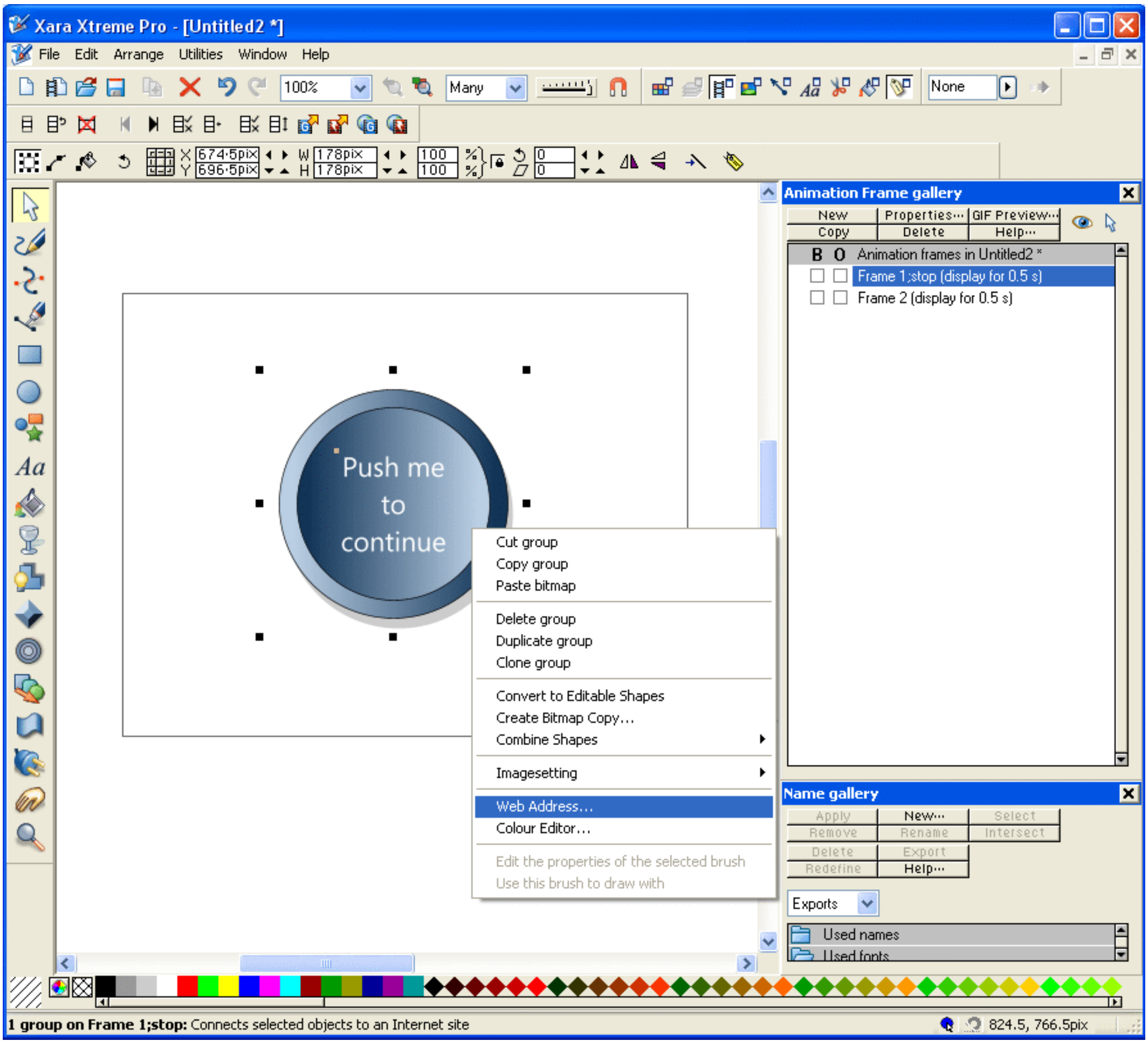
3.3 Applied to objects

Objects can be used as buttons in animations and offer the opportunity to navigate to other parts of an animation or activate different stages of an animation, or even load additional animations overlaid on the current one. This allows limited programming in Flash animations, especially when combined with the stop command applied to frame names as discussed in Section 3.2.1, which can pause an animation and wait for a user response. Commands to objects are applied using the web address dialogue when you right click on objects.

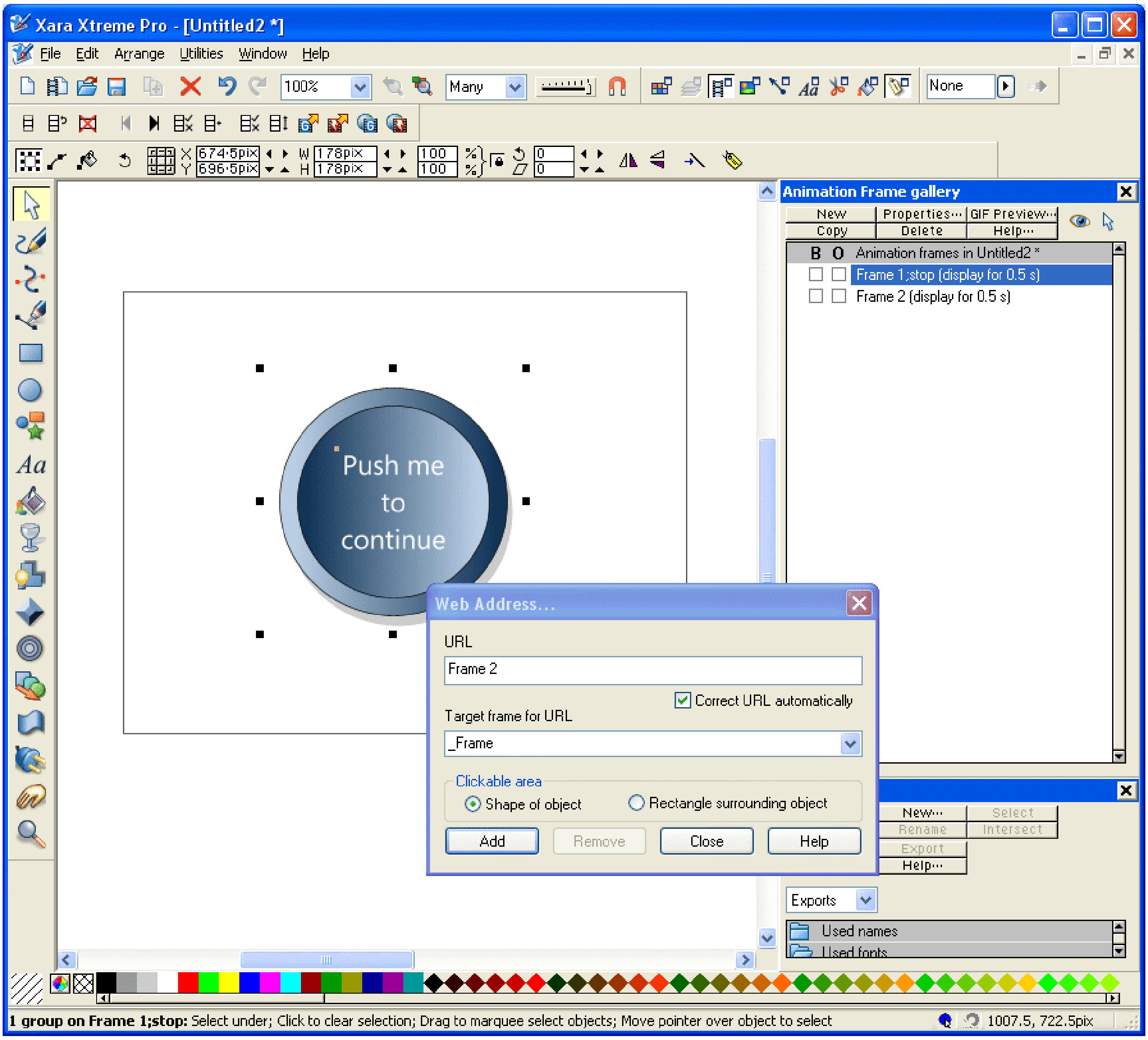
3.3.1 _frame (goto)

(See video "**Object command – goto**".)This command is used to navigate to a different part of an animation when the user clicks on a button. To redirect the animation when clicking a button, follow the procedure shown below:

1) Right-click the object that you want to be your button and select "Web address…".



2) Enter the name of the frame (excluding any commands in that frame name, such as “goto” for example) that you want to jump to when the button is pressed in the URL box. Enter “_Frame” in the “Target frame for URL” box. Click Add to assign this to the object.



B	O	Animation frames in Untitled2 *
□	□	Frame 1;stop (display for 0.5 s)
□	□	Frame 2 (display for 0.5 s)

Web Address...

URL: Frame 2

☑ Correct URL automatically

Target frame for URL: _Frame

Clickable area: ● Shape of object ○ Rectangle surrounding object

Add | Remove | Close | Help

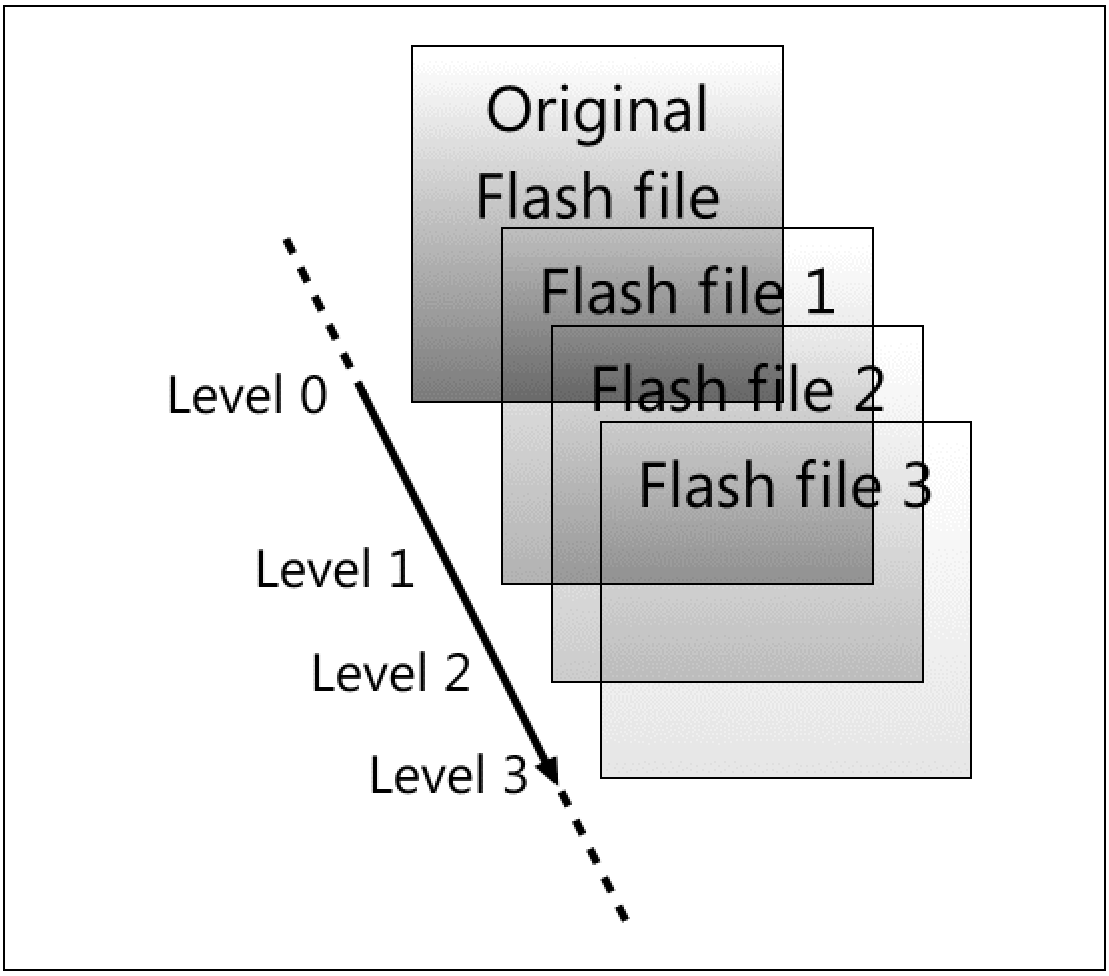
1 group on Frame 1;stop: Select under; Click to clear selection; Drag to marquee select objects; Move pointer over object to select 1007.5, 722.5pix

3) When you run the animation, it stops at Frame 1 (due to the stop command in the frame name). You are then able to press the button; clicking on it will cause the animation to begin at frame 2.

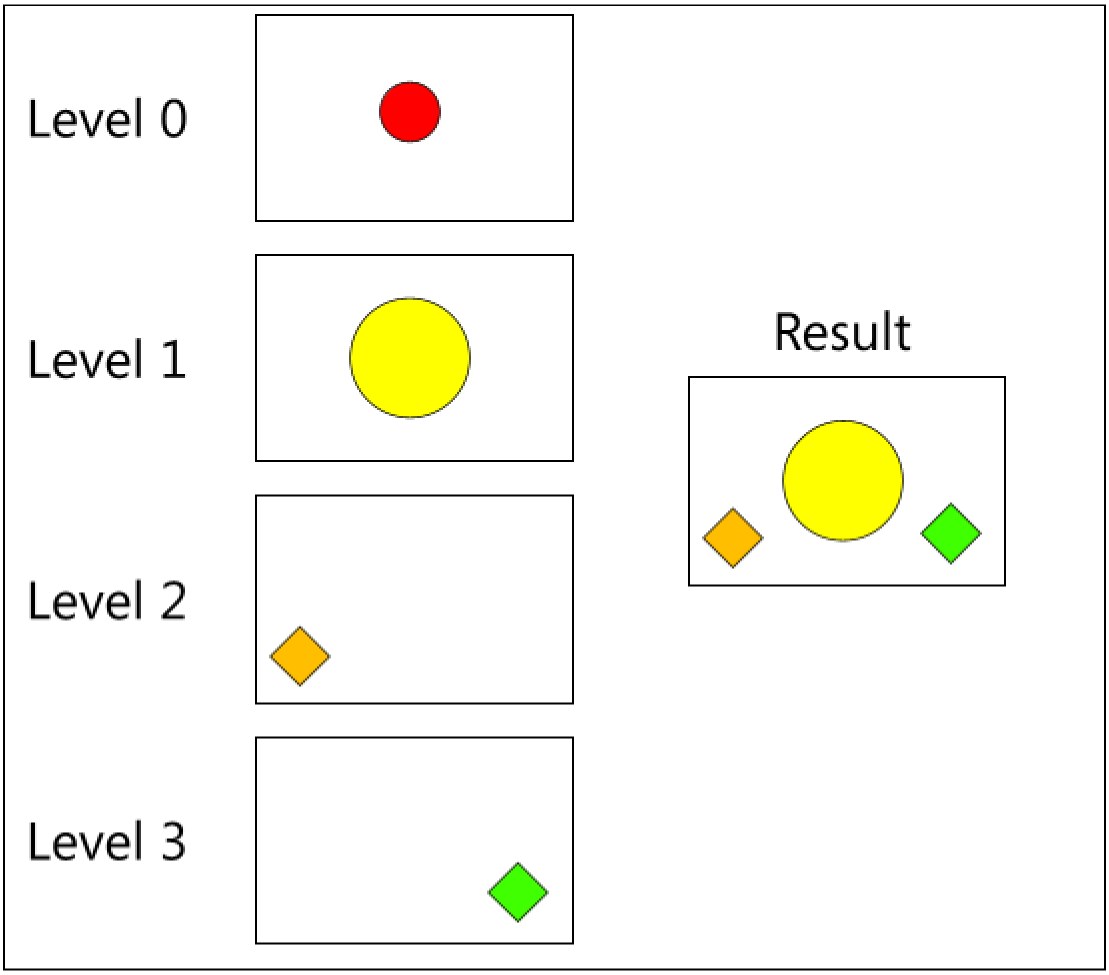
3.3.2 _level

(See video "**Object command – level**".) The _level command can be used to load other pre-made Flash animations from a core Flash file. Each animation is played independently of the one it was loaded from, and this allows a much wider range of effects to be produced in your Flash animations. It also allows more manageability in complex animations, as each File can be edited and re-exported separately.

The animations are called using the level command and, conceptually, are placed on a "level" that is above or below the level of the Flash file it was called from. The diagram below illustrates this more clearly; the base Flash file (level 0) can be used to display other Flash files on different levels (in this case all above level 0).



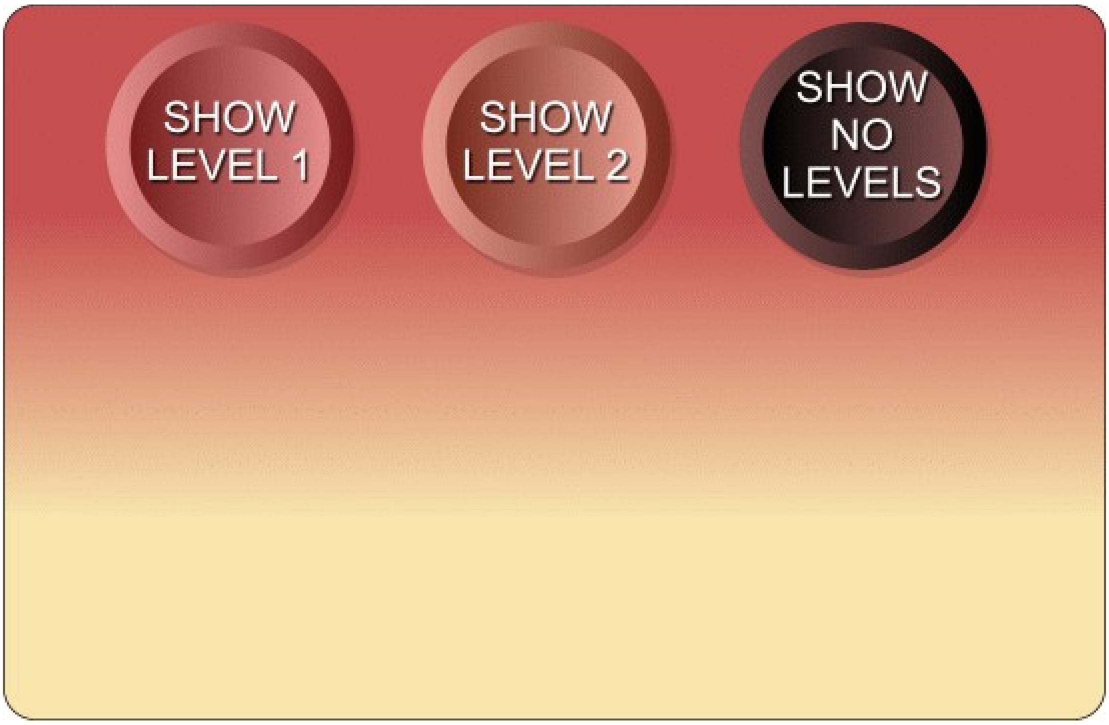
Because new Flash files are introduced almost like sheets of transparency, higher levels cover over lower levels as illustrated below. You should bear this in mind when using this feature.



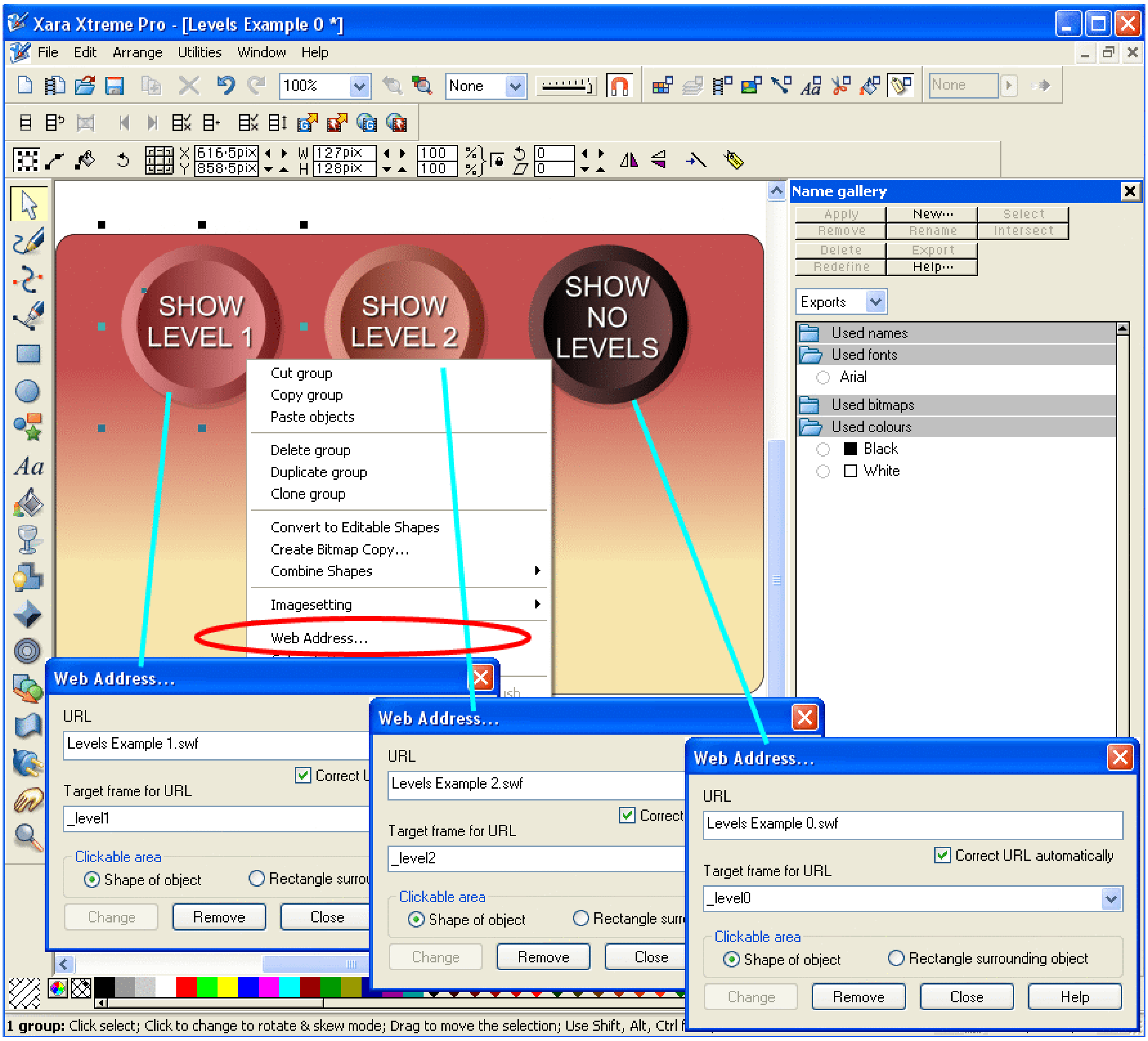
When publicly displaying Flash files that use this feature, the other Flash files must be available on your web space. You can however design Flash files that use this feature without having your own web space, and this will be discussed below in the example.

In the example, we will have in the primary Flash file a series of three buttons within a border. Two of the buttons activate two different levels (i.e. different pre-made Flash files are loaded), and the final button shuts them all off. When the levels buttons are pressed again, they will switch off the Flash animation for that level.

1. First, create the level 0, base Flash file. This can animate and then wait for a user input, but here, it's just a static image with buttons.



2. Next, right-click the first button and select "Web Address…" from the context menu, and enter the details shown below. The "URL" should be the name of the Flash file that the button will load (including the .swf extension), and the "Target frame for URL" should be set to "_levelX" where "X" represents the level it should be loaded to. Clicking the "Add" button assigns these details to the object. You don't need to close the web address dialogue when doing the same for the other buttons – just click on the new button in the Xara document and enter new details. Xara's use of modeless dialogues here makes this process more efficient. This completes the design of the base level Flash file. NOTE: setting a button (the right most one here) to load the same file on the base level (level 0) is essentially reloading and restarting the flash animation, thus turning off any other flash files opened. But this will also restart any animation if your base file is animated.



Web Address…

URL: Levels Example 1.swf

☑ Correct U

Target frame for URL: _level1

Clickable area: ● Shape of object ○ Rectangle surro

Web Address…

URL: Levels Example 2.swf

☑ Correct

Target frame for URL: _level2

Clickable area: ● Shape of object ○ Rectangle surr

Web Address…

URL: Levels Example 0.swf

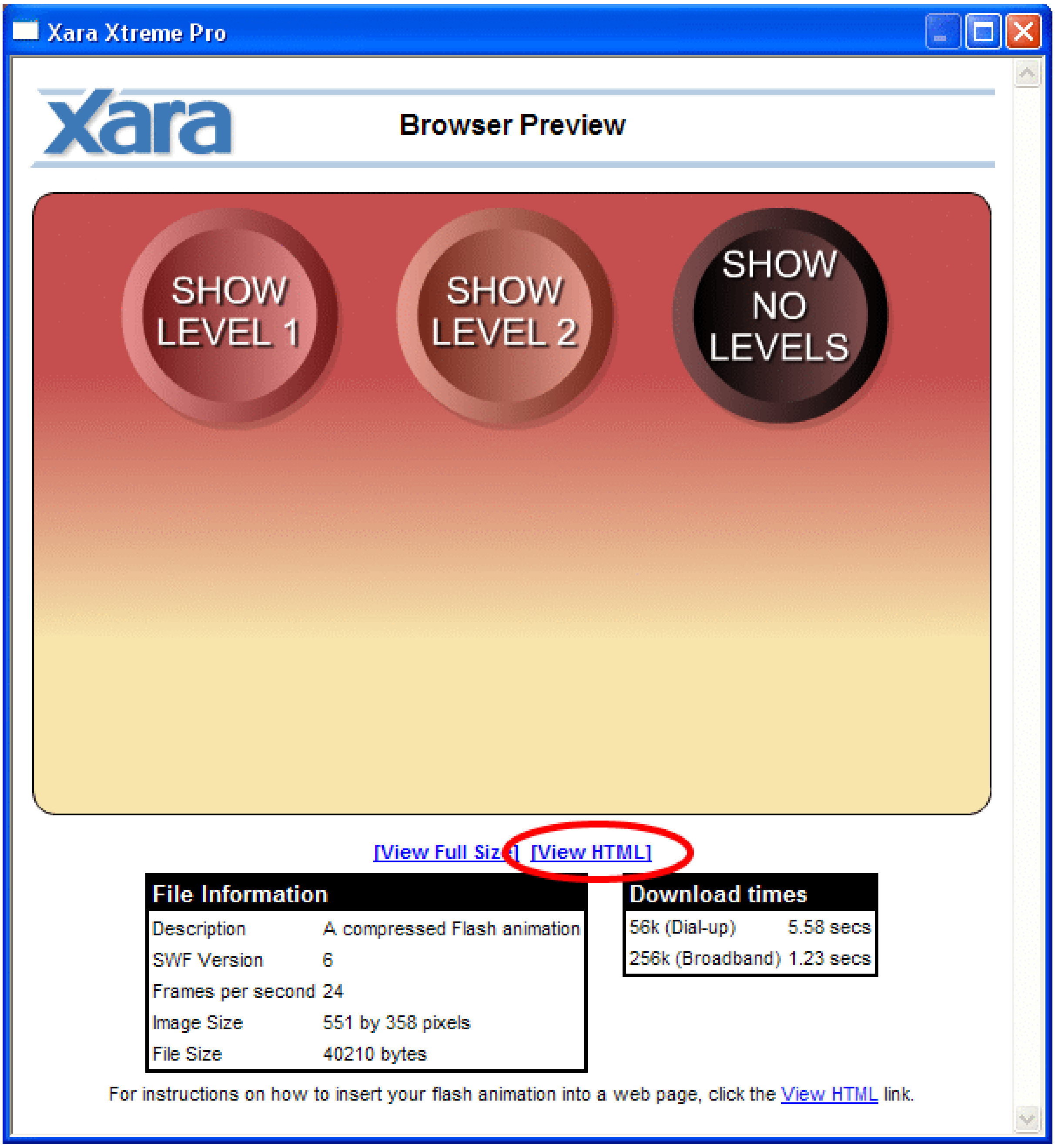
☑ Correct URL automatically

Target frame for URL: _level0

Clickable area: ● Shape of object ○ Rectangle surrounding object

1 group: Click select; Click to change to rotate & skew mode; Drag to move the selection; Use Shift, Alt, Ctrl f

3. Export the Flash file (preferably keeping the same name as the xar file for ease of editing later if needed). NOTE: all swf and html files generated in the following steps must be stored in the same directory.
4. For the base level Flash file, now preview the Flash animation, and click on the “View HTML” link.



Browser Preview

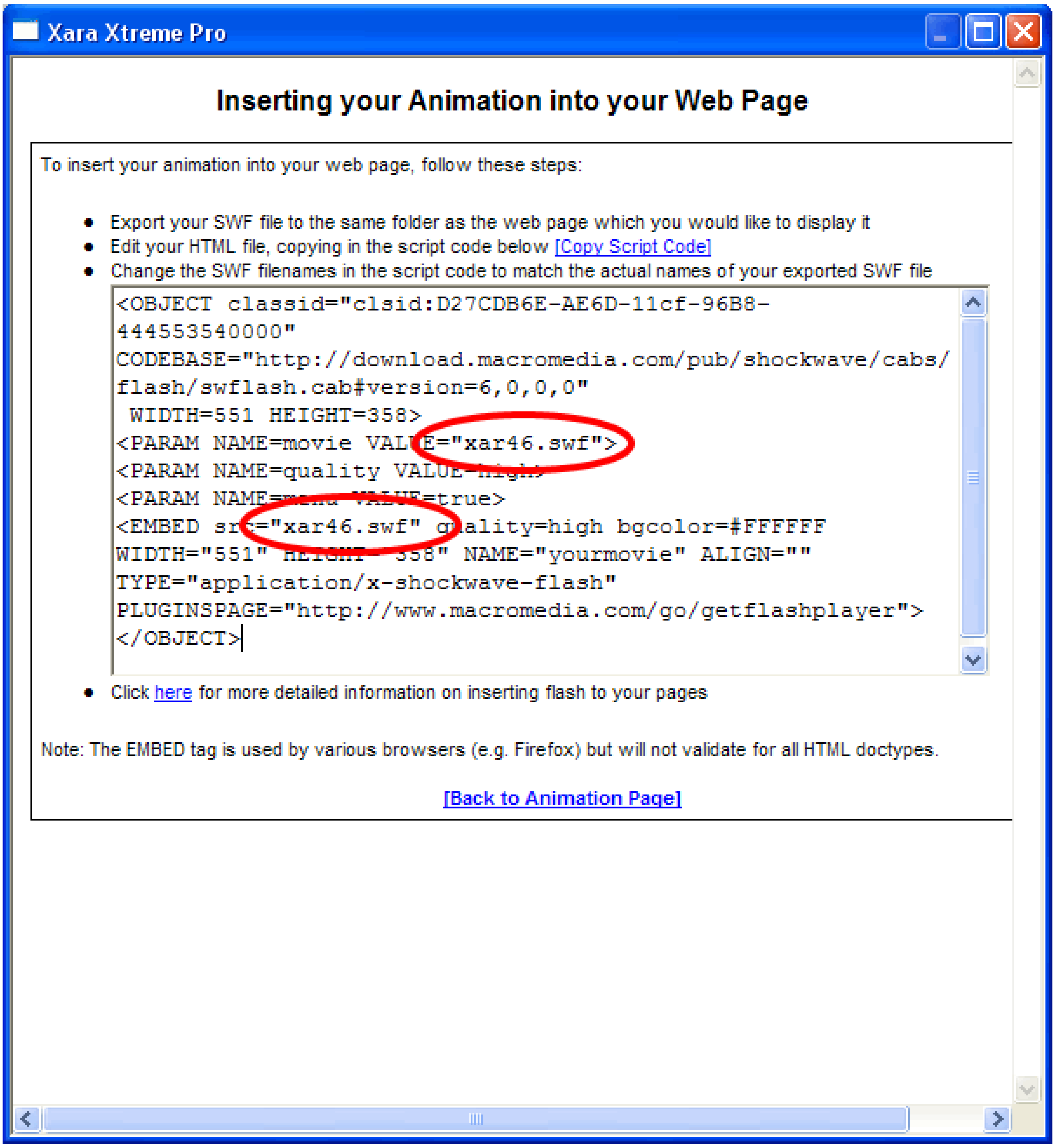
[View Full Size] [View HTML]

File Information	
Description	A compressed Flash animation
SWF Version	6
Frames per second	24
Image Size	551 by 358 pixels
File Size	40210 bytes

Download times	
56k (Dial-up)	5.58 secs
256k (Broadband)	1.23 secs

For instructions on how to insert your flash animation into a web page, click the View HTML link.

5. Copy the text presented and paste into notepad. Change the references of the temporary .xar file names to the name of the .xar file you were just working on, i.e. change “xar46.swf” to “Levels Example 0.swf”.



Inserting your Animation into your Web Page

To insert your animation into your web page, follow these steps:

- Export your SWF file to the same folder as the web page which you would like to display it
- Edit your HTML file, copying in the script code below [Copy Script Code]
- Change the SWF filenames in the script code to match the actual names of your exported SWF file

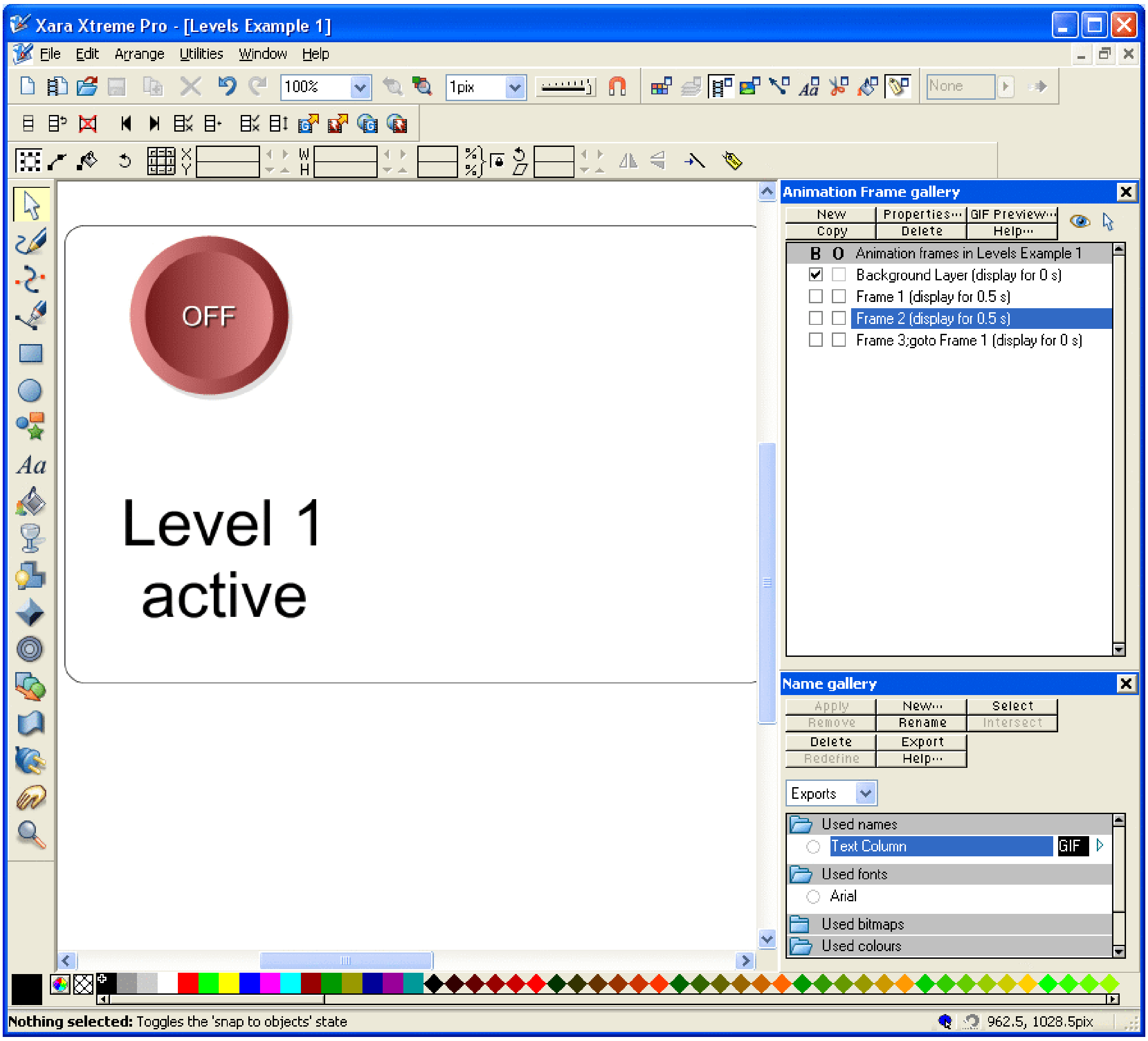
```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
CODEBASE="http://download.macromedia.com/pub/shockwave/cabs/
flash/swflash.cab#version=6,0,0,0"
 WIDTH=551 HEIGHT=358>
<PARAM NAME=movie VALUE="xar46.swf">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=menu VALUE=true>
<EMBED src="xar46.swf" quality=high bgcolor=#FFFFFF
WIDTH="551" HEIGHT="358" NAME="yourmovie" ALIGN=""
TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
</OBJECT>
```

- Click here for more detailed information on inserting flash to your pages

Note: The EMBED tag is used by various browsers (e.g. Firefox) but will not validate for all HTML doctypes.

[Back to Animation Page]

6. Save this as an .HTML file instead of a .txt file.
7. Open this in a browser (drag and drop) and your animation will show. Only the “Show no levels” button works at the moment because the other Flash files aren’t yet made, so let’s now make them.
8. The level 1 Flash file will just be a background object the same size as the level 0 background object (to ensure correct aligning), and an animating piece of text showing that level 1 is active. A new button will also cover the old one, and when this button is pressed, it will turn off the Flash file.
9. First create the objects required on the level 1 Xara file (saving it with a new name). It is recommended that you copy the needed objects from the level 0 Xara file to ensure things match up. Remember that this level will be on top of the base level Flash file and will cover anything on that original level. Hence the background is transparent.



B O Animation frames in Levels Example 1

☑ □ Background Layer (display for 0 s)

□ □ Frame 1 (display for 0.5 s)

□ □ Frame 2 (display for 0.5 s)

□ □ Frame 3;goto Frame 1 (display for 0 s)

Exports

Used names

○ Text Column GIF

Used fonts

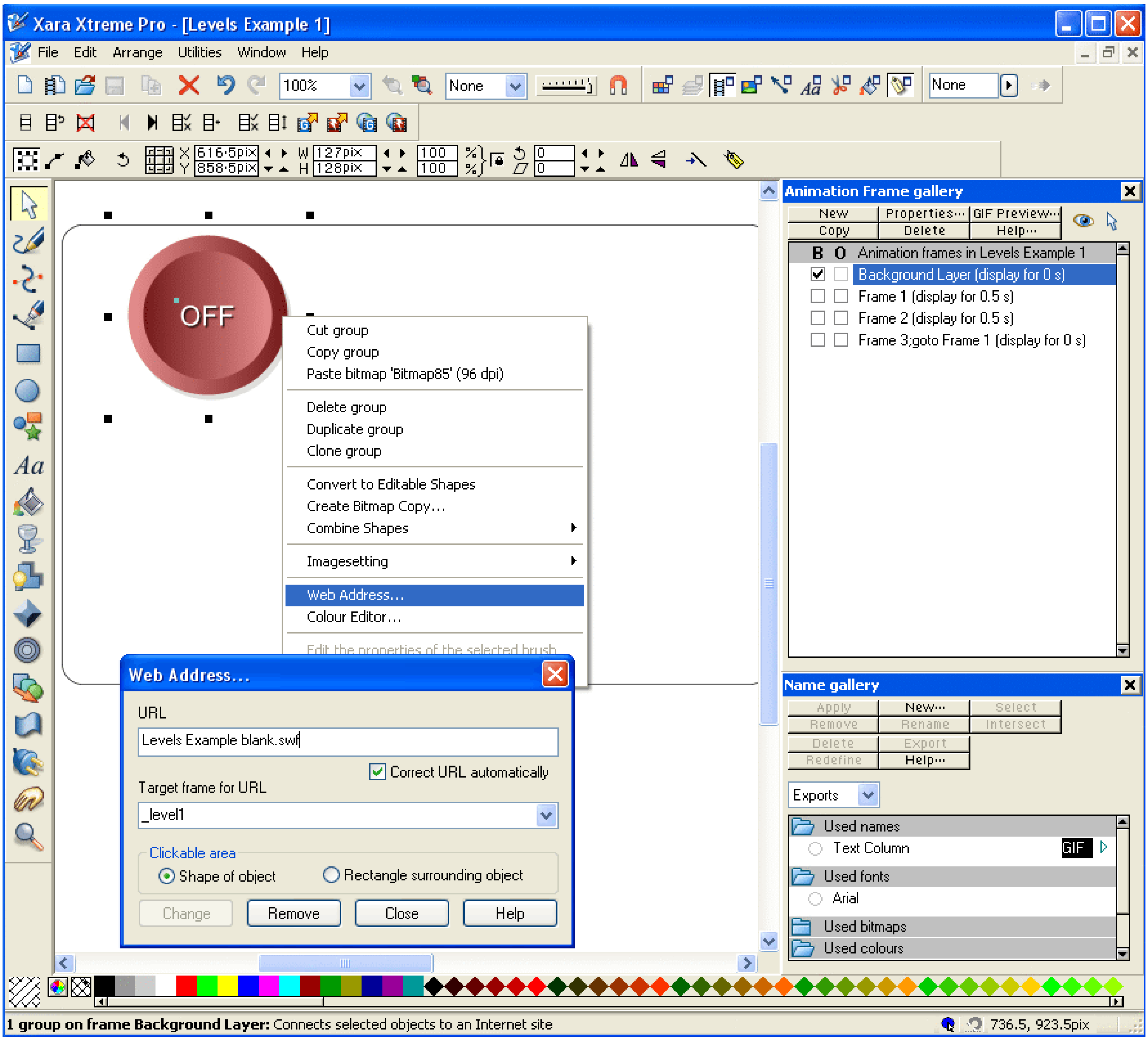
○ Arial

Used bitmaps

Used colours

Nothing selected: Toggles the 'snap to objects' state 962.5, 1028.5pix

10. To have the layer turn off when the button is pressed is made easy by Xara. Simply right click the new button and select web address again. Change the URL to be a non-existent Flash file name, e.g. “Levels Example blank.swf” or just “blank.swf” etc., and the target frame for URL should be set to _levelX where X here should be the same level we’re on, i.e. level1. In the completed Flash file, clicking on this button turns off the level.



B O Animation frames in Levels Example 1

☑ □ Background Layer (display for 0 s)

□ □ Frame 1 (display for 0.5 s)

□ □ Frame 2 (display for 0.5 s)

□ □ Frame 3;goto Frame 1 (display for 0 s)

Web Address…

URL

Levels Example blank.swf

☑ Correct URL automatically

Target frame for URL

_level1

Clickable area

● Shape of object ○ Rectangle surrounding object

1 group on frame Background Layer: Connects selected objects to an Internet site

11. Export the Flash file as “Levels Example 1.swf” and this completes the first Flash file we need. If you refresh the browser that you loaded the html file into, you’ll now see that the first button works.
12. Simply repeat steps 9-11 to create the second Flash file, and ensure all files are saved and exported.
13. Refreshing the browser should now yield a completed multi-Flash-file Flash animation with working buttons.

Any of the files called from the main one can easily be edited on their own. This is a powerful way of producing complex animations with effects that could not be achieved by just using a single Flash file.

4 Other features and techniques

4.1 Text on a curve:

(See video "**Text Tween**".) Flash supports tweening text on a curve. The curve on which the text is attached can be altered in the next key frame. To avoid errors, the curve must be set to non-visible (right-click no colour on colour line, or set line thickness to 0). Names must only be applied only to the text on the curve and NOT the curve itself.

4.2 Use of bitmaps:

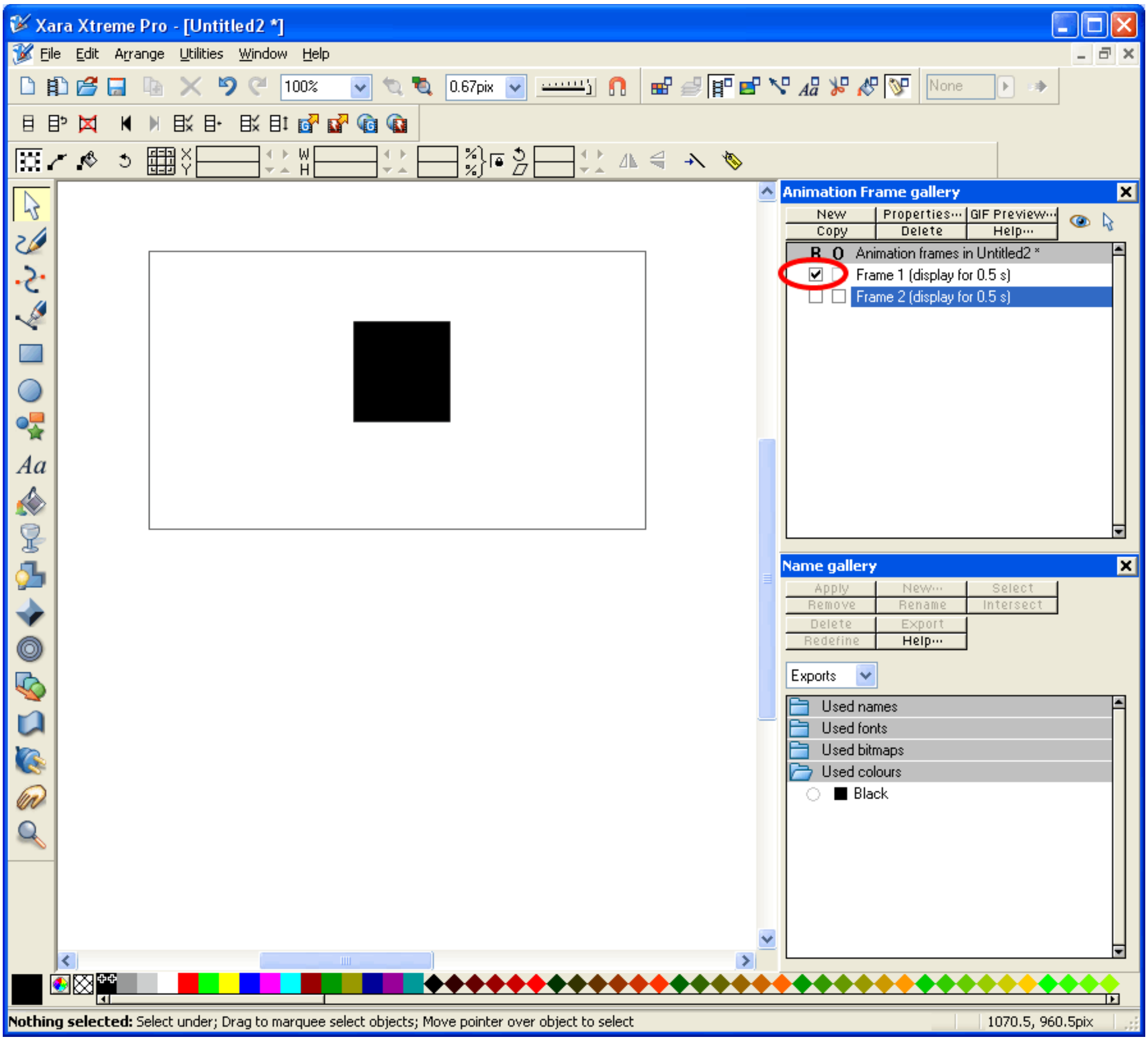
(See video "**Use of Bitmaps**".) Bitmaps (i.e. bitmap copies of objects) can be used to overcome the limitations of Flash. For example, feathering is not supported in Flash, however, if you make a bitmap copy of your feathered object, then the appearance of feathering can be retained. The disadvantage of using bitmaps in Flash however is that it increases file size proportionally to the size of the bitmaps used.

5 Other useful information

5.1 Good habits:

5.1.1 Using a background layer

It's often more efficient in an animation that isn't particularly complicated to make a background frame containing all non animating objects that appear on all frames. This is done by selecting the "B" checkbox in the frame gallery. The objects remain visible on all subsequent frames, but are only editable when the background frame is highlighted in the frame gallery.

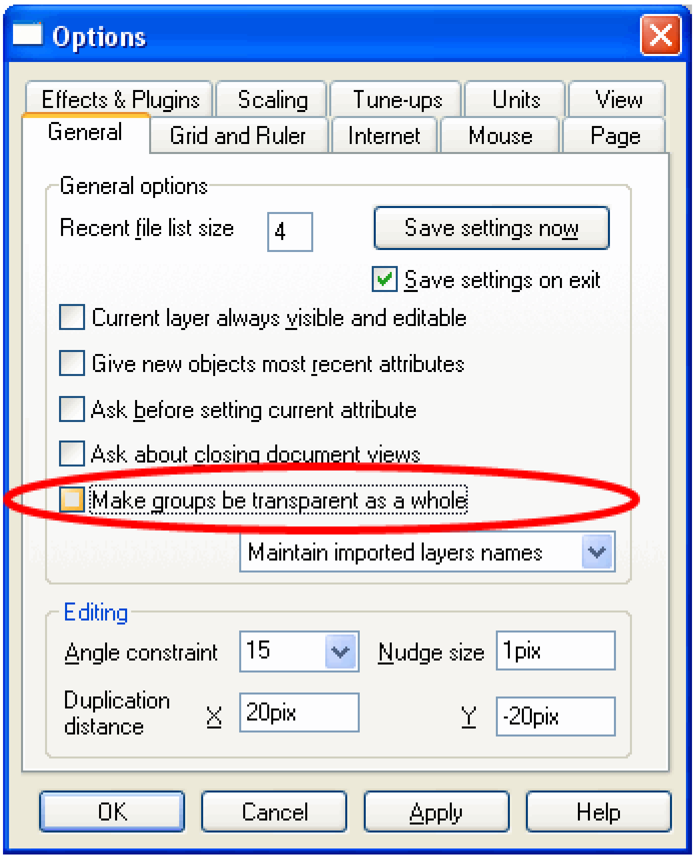


Nothing selected: Select under; Drag to marquee select objects; Move pointer over object to select

1070.5, 960.5pix

5.1.2 Other

- Check you've named objects you want to animate.
- Check animation frame before copying to create a new frame. Otherwise you could end up copying errors or omissions.
- Make sure you give separate names for each animating object.
- (See video "**Group Transparency**".) Group transparency will not work – ungroup objects first, apply transparency and then regroup and reapply name. Alternative, unselect "Make groups be transparent as a whole" to apply transparency to all individual objects in a group at once.



General options

Recent file list size: 4

☑ Save settings on exit

□ Current layer always visible and editable

□ Give new objects most recent attributes

□ Ask before setting current attribute

□ Ask about closing document views

□ Make groups be transparent as a whole

Maintain imported layers names

Editing

Angle constraint: 15

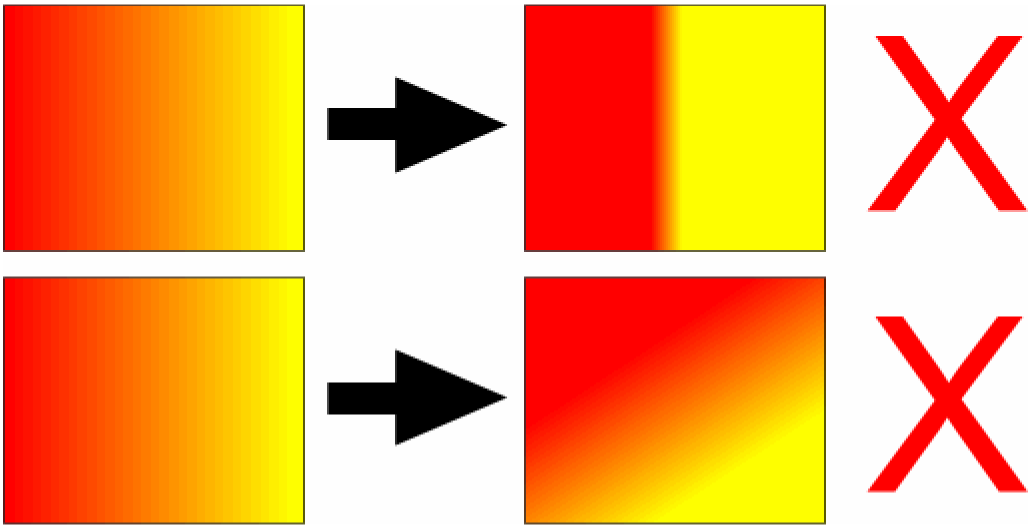
Nudge size: 1pix

Duplication distance X: 20pix

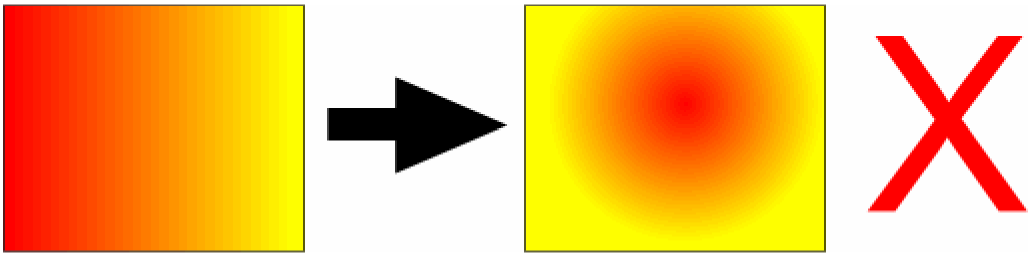
Y: -20pix

5.2 What Flash doesn't support:

- Simultaneous use of complex fills and transparency e.g. linear fill and linear transparency – you can use only one.
- Change of position of handles of complex fills or transparencies e.g., you can't have an object fade from yellow to red subtly in frame 1, and then more abruptly in frame 2 (see image below).



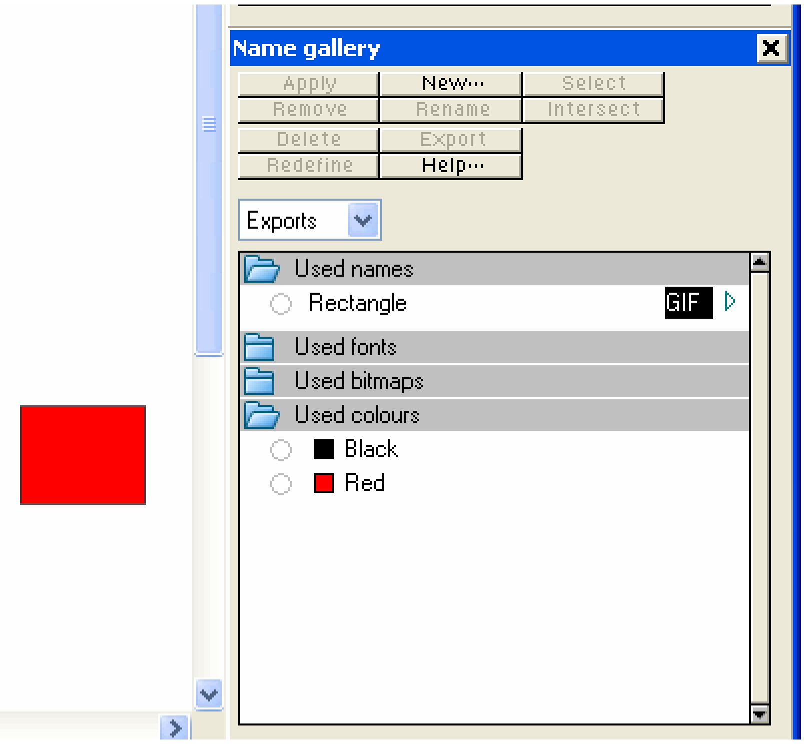
- Change of fill or transparency type between frames, e.g. from linear to circular.



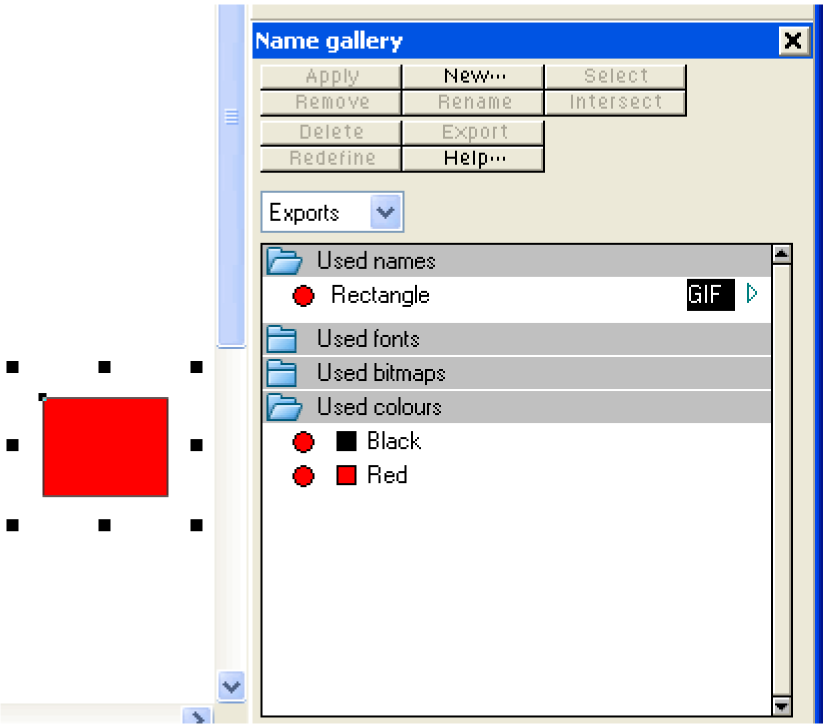
- Other more complex fills or transparencies, such as conical etc.
- Any change that causes an effective morph – i.e. a change contradicting 6 rules is banned.
- For others, see Xara's help file.

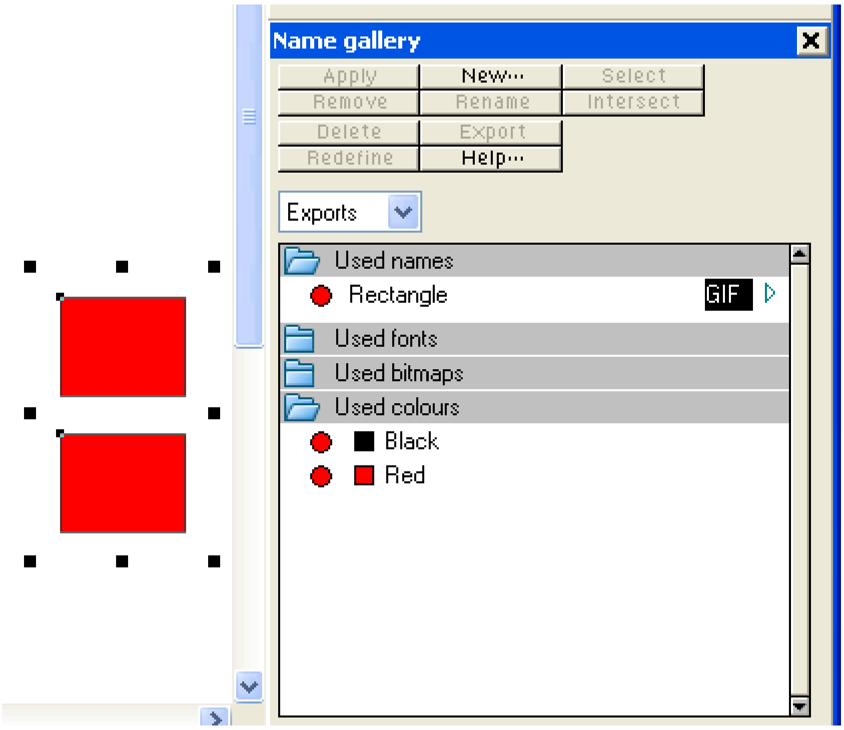
5.3 Name gallery

(See video "**Name Gallery**".) With no objects selected, or unnamed objects selected, any names in the name gallery show an empty circle indicating that objects with that name are not selected.

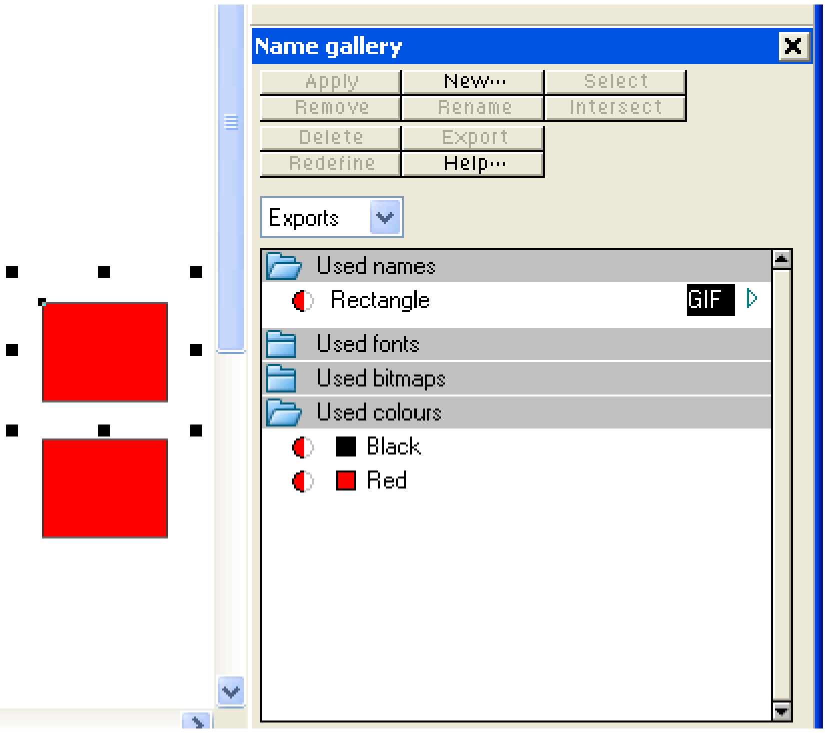


When all objects with a particular name are selected, the name is shown with a full circle.

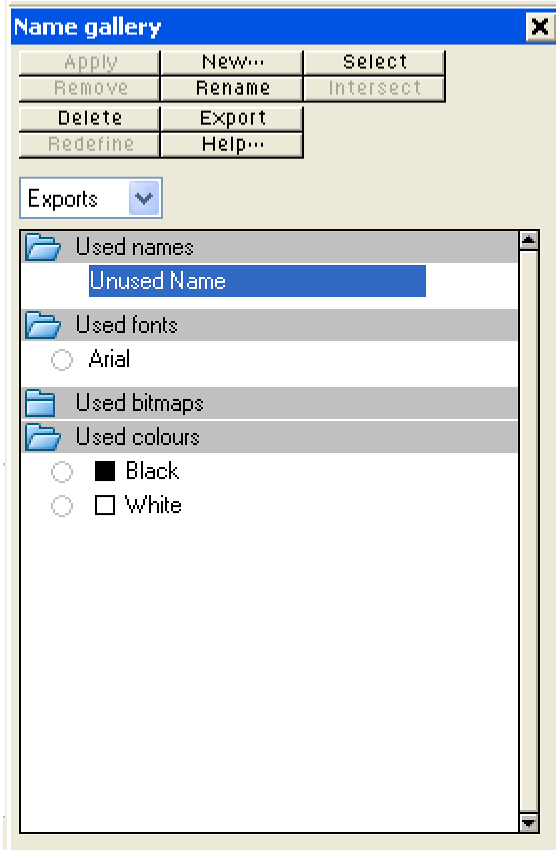




When some, but not all, objects with a particular name are selected, then the circle is half-filled.

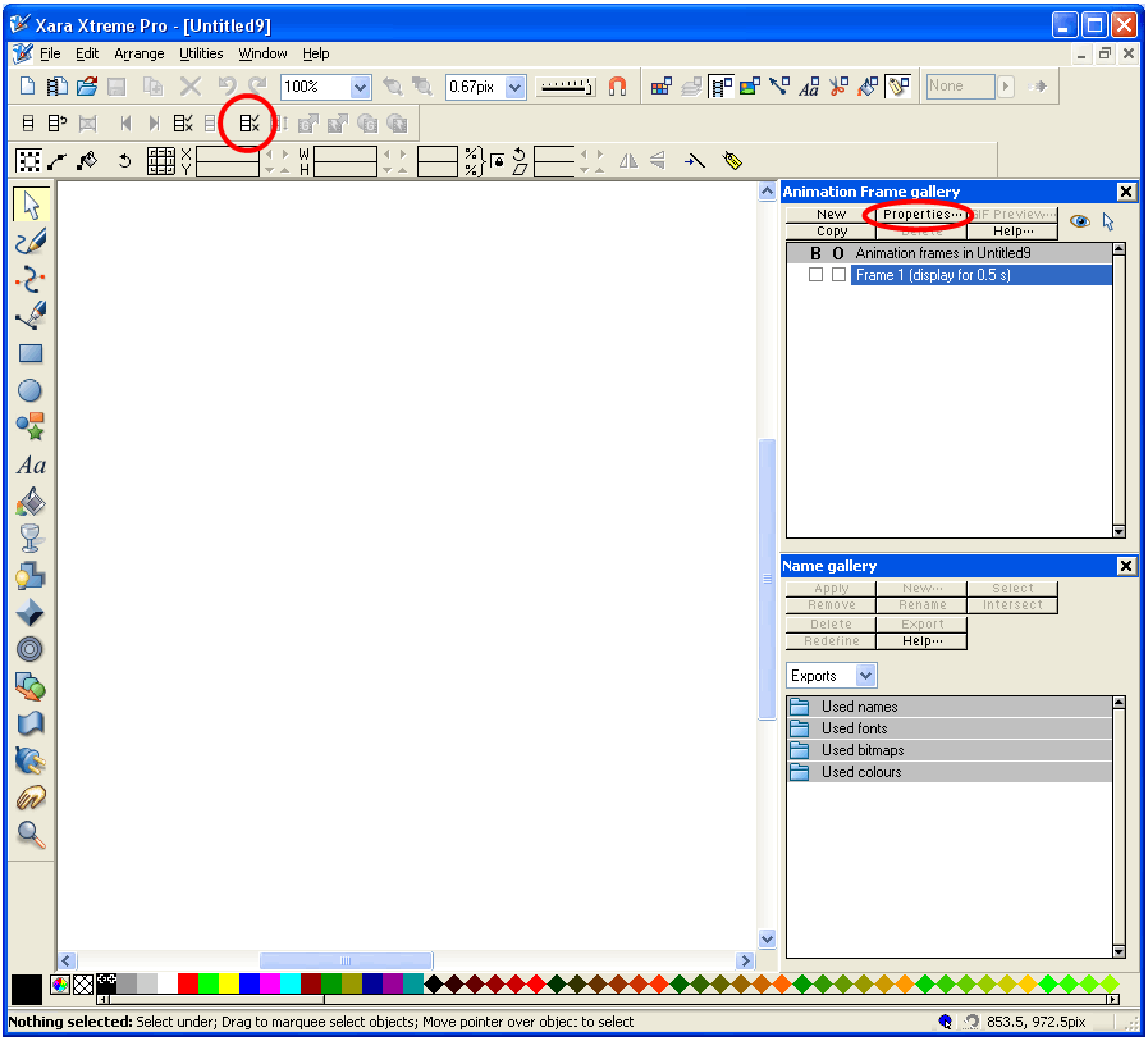


When a name exists but no objects have it assigned, there is no circle present at all.



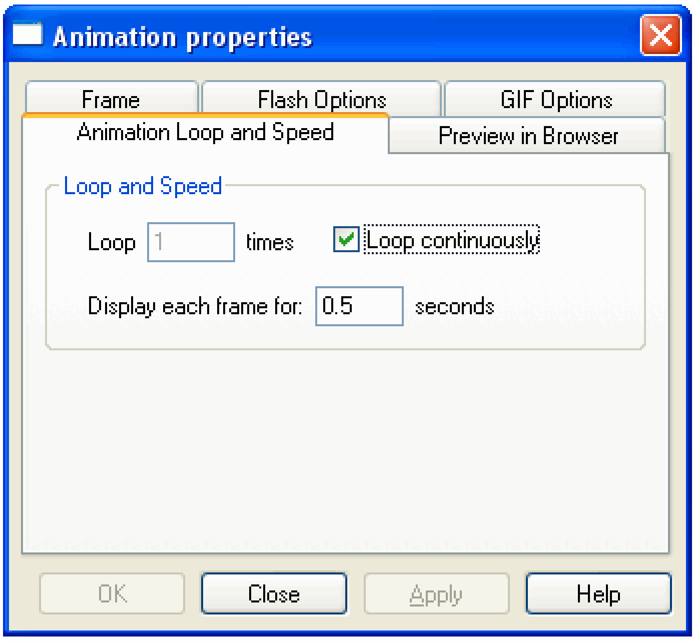
5.4 Flash properties dialogue

The Flash properties dialogue can be opened by clicking one of the highlighted buttons shown below.



Nothing selected: Select under; Drag to marquee select objects; Move pointer over object to select

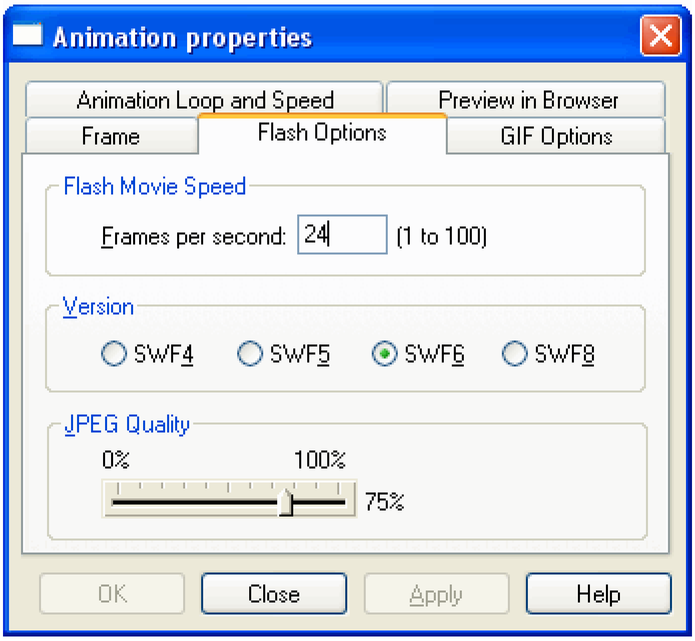
The dialogue box allows you to determine a number of different properties for your Flash animation, including frame timings, and frame rate. It is mostly self explanatory, but Xara's help file covers each option.



Loop and Speed

Loop 1 times ☑ Loop continuously

Display each frame for: 0.5 seconds



Flash Movie Speed

Frames per second: 24 (1 to 100)

Version

○ SWF4 ○ SWF5 ● SWF6 ○ SWF8

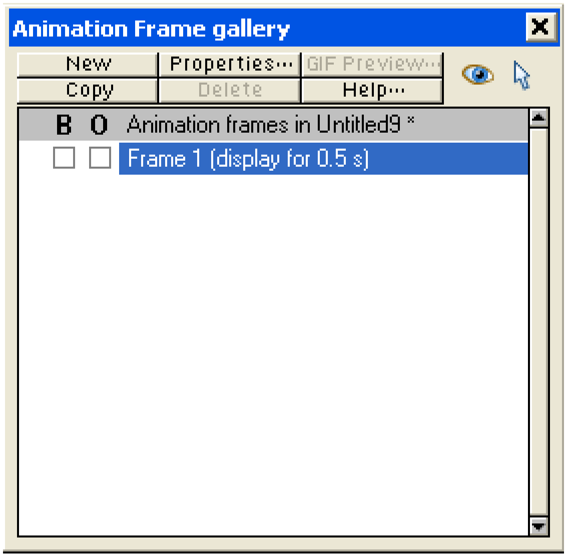
JPEG Quality

0% 100% 75%

5.5 Tips

- Set the final frame of an animation to display for 0 seconds to prevent any delay at the end when the animation is to repeat.

- The time shown on a frame in the frame gallery is the time it takes to get from that frame to how things look in the next.



- Objects can be grouped or joined to be treated as a single object for naming purposes. This prevents the need to name each object in the group separately, and can save a lot of time.

6 Videos

Videos associated with this tutorial may possibly be found in future on YouTube under user XaraXhris. Search for Xara clips.

Xhris