Drag A Sprite

Introduction

The DragASprite module provides SWI calls with which you can make the pointer drag a sprite around the screen. Since not all users will prefer this effect to dragging an outline - whether for aesthetics or performance - there is a bit in the CMOS RAM used to indicate their preference. (See the chapter entitled Non-volatile memory (CMOS RAM).) You should examine that bit before using this module; if it shows that the user would prefer to drag outlines, oblige them!

To drag a sprite:

1. Prepare a sprite to be dragged (this may be trivial, as the application may have a suitable sprite already to hand).
2. Call the SWI *SWI DragASprite_Start (on page 2)*. This takes a copy of your sprite - so you can dispose of your copy whenever you like - and then starts a Wimp drag.
3. When the Wimp sends you an indication that your drag has finished, you should call the SWI *SWI DragASprite_Stop (on page 5)* to release the workspace used for the drag.

SWI calls

<div align="right">

DragASprite_Start
(SWI &42400)

</div>

Start a sprite drag operation

On entry

R0 = flags :

Bit(s)	Meaning
0 - 1	Horizontal location of sprite in box:

	Bit(s)	Meaning
	00	left
	01	centre
	10	right

Bit(s)	Meaning
2 - 3	Vertical location of sprite in box:

	Bit(s)	Meaning
	00	bottom
	01	centre
	10	top

Bit(s)	Meaning
4 - 5	Drag bounding box is:

	Bit(s)	Meaning
	00	whole screen
	01	display area of window that the pointer's over
	10	specified in block pointed to by R4

Bit(s)	Meaning
6	Bounding box applies to:

	Bit(s)	Meaning
	0	the box
	1	the pointer

Bit(s)	Meaning
7	Control of drop-shadow:

	Bit(s)	Meaning
	0	don't do a drop-shadow
	1	make a drop shadow when copying the sprite

Bit(s)	Meaning
8	Control of translucency (just cross-hatching in current version):

	Bit(s)	Meaning
	0	dragged sprite will be translucent
	1	dragged sprite will be solid

Bit(s)	Meaning
9	Control of sprite centering:

	Bit(s)	Meaning
	0	sprite position is determined by bits 0 - 3
	1	sprite will be centered on the pointer

Bit(s)	Meaning
10 - 31	Reserved for future use - should be set to 0

R1 = sprite area holding sprite:

Value	Meaning
0	system sprite area
1	wimp sprite area
Other	address of sprite area

R2 = pointer to sprite name
R3 = pointer to 16-byte block containing box
R4 = pointer to optional 16-byte block containing bounding box (see flags)

On exit

R0 - R4 preserved

Interrupts

Interrupts are undefined
Fast interrupts are enabled

Processor mode

Processor is in SVC mode

Re-entrancy

Not defined

Use

This call starts dragging a sprite. The sprite you supply is copied, so there is no problem if you dispose of your copy of the sprite. If there is insufficient memory available to start the drag, the call reverts to a normal drag of a dotted outline.

The blocks pointed to by R3 and - optionally - R4 have the following format:

Offset	Name	Use
+0	x-low	
+4	y-low	bottom-left (x-low, y-low) is inclusive
+8	x-high	top-right (x-high, y-high) is exclusive
+12	y-high	

Related SWIs

SWI DragASprite_Stop (on page 5)

DragASprite_Stop
(SWI &42401)

Terminates any current drag operation, and releases workspace

On entry

None

On exit

None

Interrupts

Interrupts are undefined
Fast interrupts are enabled

Processor mode

Processor is in SVC mode

Re-entrancy

Not defined

Use

This call terminates any current drag operation, and releases any
workspace claimed by the DragASprite module to do a drag. You should
make this call when your application receives the User_Drag_Box reason
code from SWI Wimp_Poll during a drag.

Related SWIs

SWI DragASprite_Start (on page 2)

Document information

Maintainer(s):	RISCOS Ltd <developer@riscos.com>
History:	**Revision Date Author Changes**
	1 ROL Initial version
Disclaimer:	Copyright © Pace Micro Technology plc, 2001.
	Portions copyright © RISCOS Ltd, 2001-2004.
	Published by RISCOS Limited.
	No part of this publication may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, or stored in any retrieval system of any nature, without the written permission of the copyright holder and the publisher, application for which shall be made to the publisher.