Point   structure

#include <Types.h>

typedef struct **Point** {	Size	Offset	Description
short v;	2	0	Vertical coordinate
short h;	2	2	Horizontal coordinate
} **Point**; 4			

typedef Point **Cell**;   (used in List Manager calls)
typedef Point ***PointPtr**;

Notes: Use **SetPt** to initialize a **Point** data structure, or simply assign values to the members directly.

Note: This structure is 32-bits with the v value in the hi word and the h value in the low word. In many cases, you may be able to optimize by using 32-bit register operations.   You may also want to coerce a long (eg, the value returned by **DeltaPoint**) into a Point for easier handling.

The **Point** (a.k.a. **Cell**) structure is used in calls to:

AddPt	**GrowWindow**	**LocalToGlobal**	**SFGetFile**
DeltaPoint	**LAddToCell**	**LRect**	**SFPGetFile**
DIBadMount	**LCellSize**	**LSetCell**	**SFPPutFile**
DragControl	**LCellSize**	**LSetSelect**	**SFPutFile**
DragGrayRgn	**LClick**	**MapPt**	**ShieldCursor**
DragWindow	**LClrCell**	**MenuSelect**	**StdLine**
EqualPt	**LDraw**	**PinRect**	**SubPt**
FindControl	**LFind**	**Pt2Rect**	**TEClick**
FindDItem	**LGetCell**	**PtInRect**	**TestControl**
FindWindow	**LGetSelect**	**PtInRgn**	**TrackBox**
GetMouse	**LLastClick**	**PtToAngle**	**TrackControl**
GetPen	**LNew**	**ScalePt**	**TrackGoAway**
GlobalToLocal	**LNextCell**	**SetPt**	

In some cases, a **Point** is used to specify a height and width (as in the cellSize field of a ListRec structure) or a vertical and horizontal distance (as in **SubPt**). In these cases, the .h field is always a width (or horizontal delta) and .v is always a height (or vertical delta).