Subject: Re: moving to objects / IDL objects and object graphics
Posted by Rick Towler on Tue, 09 Oct 2007 16:09:24 GMT
View Forum Message <> Reply to Message

markb77 wrote:

I would suggest creating your MBgrMovie object as a subclass of IDLgrImage. Keep it simple. All you really need to add are the properties you have mentioned, a timer as Robbie mentioned, a pointer to the movie data, and some methods to make it all work (stop, start, fwd, rev, go to, etc).

I personally would separate the GUI from the movie object so I would only include one hidden base widget which would handle the timer events. Your object's timer event will do two things. Update your DATA property with "this" frames image data and optionally call a callback. Create a property that contains the name of the callback that your hidden timer event calls when your movie object needs to draw the next frame (check out call_function and call_procedure). Construct your callback such that you pass your movie object reference as well as a pointer to "this" frames image data. Including the pointer to the image data would allow you to use it in direct graphics applications as well.

> But could this be integrated with IDL object graphics? For instance,
> would the IDLgrView know what to do if I were to Add my Movie object
> to it? And would the IDLgrWindow know how to draw it?

Your object will be a subclass of IDLgrWindow so all you need to do is add it into your object hierarchy. In this case I don't think you need to override the DRAW method as David suggests. Your movie is ruled by the timer. Just get the proper frame into the image object at the right time and let the callback routine know. It will be the application's responsibility to call the draw method for the view.

At some point you may want to consider more elegant methods to handle the image data. Holding the entire movie in RAM may not always be an option.

-Rick