
Subject: Re: Top 10 IDL Requests

Posted by [Mark Hadfield](#) on Mon, 31 Jul 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

> "Kristian Kjaer" <Kristian.Kjaer@Risoe.Dk> wrote in message
news:3981DA40.F3BC8FC9@Risoe.Dk...

>

> A print button (and an equivalent cmd-line command) which would work
> on (direct) graphics already rendered to the screen (using std. direct
> graphics commands) would null _the_ major quirk in IDL, IMHO.

And how would it be done?

Once a direct graphics command has sent output to an output device, the only "memory" IDL has of that command is the changed state of the output device. At that point the system (or the user) has two ways of recreating the output to a different device:

1. Switch devices & re-issue the same commands
2. Read the output back off the device and send it to the new device.

Number 1 is the method that users normally employ. It's hard to automate because the graphics commands may well have change the state of the system in all sorts of unknown ways. This is the method employed by David Fanning's direct graphics objects. It's a brilliant idea but it has a major limitation: it requires the user to bundle the graphics command in a single routine that is "well-behaved", i.e. it can be run repeatedly & is responsible for recreating all its data every time it is called.

Number 2 is already available for graphics windows via TVRD or the "copy to clipboard" functionality on Windows. The graphics output is read back from the output device as a raster image. This is pretty much useless when redirected to a device with a different resolution.

The "directness" of IDL direct graphics is one of the main reasons for its speed but also the source of fundamental limitations. It's because of these fundamental limitations that RSI invented object graphics.

Mark Hadfield

m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>

National Institute for Water and Atmospheric Research

PO Box 14-901, Wellington, New Zealand
