
Subject: Re: programmatically detect problems in idl code
Posted by [Michael Wallace](#) on Mon, 01 Aug 2005 17:57:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

If I understand what you're asking, you'd like an appropriate exit status returned to the operating system such that you can check that number and determine if you IDL code executed successfully. Is this what you're after?

By default, IDL will always return an exit status of 0 regardless of whether IDL was successful or not. You can set your own exit status numbers with the exit command. For example,

```
exit, STATUS = 1
```

will immediately exit your program and the operating system will see a value of 1 returned from the process. You can simply put a line like this at the end of your main program. If all of your code was successful, the OS will see the value 1. If there was an error anywhere along the way, you'll see the value 0.

If you want to get fancier, you can use the `on_error`, `on_ioerror`, and `catch` commands as mentioned in a previous message. These commands will enable you to catch certain errors and then exit your program with a custom status code for each error type. One word of warning however -- make sure you only put the exit codes in your main program and not in any of your subroutines. Doing so can lead to code that is very difficult to debug and can be very error prone. It's just good practice to throw the error back out to the main level before exiting.

-Mike
