
Subject: Resolving Built-ins and FORWARD_FUNCTION

Posted by [JD Smith](#) on Tue, 09 Dec 2003 00:43:05 GMT

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

Sent to RSI:

```
=====
=====
```

Using FORWARD_FUNCTION creates an unresolved stub in the routine list, even for built-in routines. E.g., in the NasaLib WRITEFITS you find :

```
FORWARD_FUNCTION FILE_SEARCH ;For pre-V5.5 compatibility
```

This causes problems for RESOLVE_ALL, which dutifully attempts to compile the listed built-in:

```
IDL> .run 'writefits'
% Compiled module: WRITEFITS.
IDL> print,routine_info(/FUNCTIONS,/UNRESOLVED)
FILE_SEARCH HEADFITS SXPAR
IDL> resolve_all
% Compiled module: RESOLVE_ALL.
% Compiled module: CHECK_FITS.
% Compiled module: MKHDR.
% Compiled module: SXADDPAR.
% Compiled module: SXDELPAR.
% Attempt to call undefined procedure/function: 'FILE_SEARCH'.
% Error occurred at: RESOLVE_ALL_BODY 280
  /usr/local/rsi/idl_6.0/lib/resolve_all.pro
%          RESOLVE_ALL 332
  /usr/local/rsi/idl_6.0/lib/resolve_all.pro
%          $MAIN$
% Execution halted at: $MAIN$
```

IDL could either check for built-in's being used in FORWARD_FUNCTION, or RESOLVE_ROUTINE could do the same, or FORWARD_FUNCTION functions could be removed from the list once they are encountered in the file. Since you can't override a built-in command (like FILE_SEARCH) with any amount of !PATH fiddling, it makes sense not to put built-ins on the unresolved list via FORWARD_FUNCTION.

```
=====
=====
```

Also, does anyone know what a SAV file run in the IDLVM does with a statement like:

```
source=routine_info('MyPro',/SOURCE)
```

I use these types of constructs to locate data bundled with my source distribution, and I want it to work with the IDLVM too. Since the VM technically doesn't do any compiling of files, I presume it might not do any path searching for file source either, in which case I'd have to come up with something different.

Thanks,

JD
