Subject: Re: user defined idl_catalog.xml
Posted by JD Smith on Fri, 22 Sep 2006 16:52:16 GMT
View Forum Message <> Reply to Message

On Fri, 22 Sep 2006 11:01:58 -0400, sebinjapan wrote:

> Hello,
>
> I use idldoc package developed by M. Galloy and the emacs mode IDLWAVE
> developed by JD Smith.
> - The first one allowed me to create a nice documentation of my routine
> that can be browsed with the idl_assistant.
> - The second one has the ability to automatically load the online help for
> the nearby IDL routine [M-?]. To do this, it uses the file idl_catalog.xml
> provided with IDL (in .../idl/help/online_help).
>
> It would be great if IDLWAVE was able to load the idl_assistant pages I
> created with idldoc when looking at my files in emacs. To do this I guess
> it would be necessary to create an idl_catalog.xml for my own files. Does
> anyone know how to do that?

Interesting... any screenshots of an example? Don't you need an "ADP"
file as well (the IDL_Assistant wants one).

Right now, IDLWAVE simply parses that XML file, and then makes a LISP
equivalent in ~/.idlwave/idl_xml_rinfo.el for quicker re-load.

If you provide source code along with your package, I have thought of
creating a special tag in the documentation header, along the lines
of:

;; %%%HTML-HELP%%% /path/to/html/help/file.html

or some such, which can be loaded into the assistant if found. This
doesn't help if you distribute binary flavors of your routines,
however. The XML file itself is fairly straightforward to duplicate.
Aside from special cases like aliases (for those space-saving
documentation entries like OPENU/OPENR/OPENW), and system variables,
etc., the format is pretty simple:

```
<ROUTINE name="WV_APPLET" link="WV_APPLET.html">
  <SYNTAX name="WV_APPLET [, Input]" type="pro" />
  <ARGUMENT name="Input" link="WV_APPLET.html#wp1004581" />
  <KEYWORD name="ARRAY" link="WV_APPLET.html#wp1004583" />
  <KEYWORD name="GROUP_LEADER" link="WV_APPLET.html#wp1004841" />
  <KEYWORD name="NO_SPLASH" link="WV_APPLET.html#wp1004848" />
  <KEYWORD name="TOOLS" link="WV_APPLET.html#wp1004856" />
  <KEYWORD name="WAVELETS" link="WV_APPLET.html#wp1005269" />
```

</ROUTINE>

i.e., name the routine, give the link (here assumed to be beneath $IDL_PATH/help/online_help), give syntax with (important) type="pro" or type="func", list the arguments and their anchor links (if they have separate links in the file), and similarly with keywords.

Class methods are slightly different -- see the xml file.

However, even if you create such a file, there is no simple way to append the data to the one shipped with IDL. IDLWAVE assumes the two sources of routine info are separate (library catalogs vs. system routine info).

If IDLDOC simply generates a webpage without lots of internal links (for keywords, etc.), it would be simpler to just associate that html file somehow with the source code through some markup convention. Then rather than using "in source" help for library files, it could call the assistant for display. We'd also need a convention for where the IDLDOC html files are stored relative to the .pro file, e.g. just specify relative directories:

;; %%%HTML-HELP%%% ../help/myroutine.html

Perhaps Michael could overload the IDLDOC syntax to include something like this, assuming that fits in logically. If it were trivial to unambiguously recognize a block of IDLDOC markup (which it doesn't appear to be as of IDLDOC 2.0), IDLWAVE (or, rather, idlwave_catalog, the perl script which created library catalog files) could look for a specific tag, like:

; @help_location ../help/myroutine.html

save this path to the HTML file along with other information on the routine (for simplicity, likely the next routine it finds in the file), and then direct all help requests for that routine to the IDL Assistant (or other configured web browser).

That would be more straightforward than going the full XML route. What that doesn't give you is the ability to jump right to the documentation of a given keyword. E.g. try:

IDL> widget_control, b, SCR_XSIZE=[M-?]

I don't know if IDLDOC would ever provide that granularity.

JD
