
Subject: Re: IDLWAVE 4.15 Release -- idlwave.org
Posted by [Carsten Dominik](#) on Thu, 12 Sep 2002 07:32:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

JD had done fantastic work. Thanks a lot, this release contains a huge amount of goodies.

I am going to check in the the sources into the XEmacs tree, so within a few weeks, XEmacs users should be able to get this new release directly from the XEmacs package depositories (through the Package Management system of XEmacs).

- Carsten

>>>> > "JS" == JD Smith <jdsmith@as.arizona.edu> writes:

JS> IDLWAVE 4.15 -- <http://idlwave.org>
JS> A hot new release of IDLWAVE is available for those of you addicted to
JS> new features, bug fixes, and the relentless march of ever-burgeoning
JS> version numbers. Following the time-honored tradition of small even and
JS> large odd releases, this version of IDLWAVE serves up a bevy of choice
JS> morsels:

JS> =====> New features: <=====

JS> - A special loadable module, `idlw-roprompt' allows Emacs 21 users to
JS> make the "IDL>" prompt read-only (like it is in an XTerm). Emacs 21
JS> also brings many new features to the shell, including toolbars, the
JS> ability to middle-click on old input to have it automatically copied
JS> to the command line, colored prompt/input, etc. Highly recommended.

JS> - When completing methods or method-keywords, or invoking help or
JS> routine info on objects in the shell, IDLWAVE now automatically
JS> queries IDL for the class of that object, e.g.:

```
IDL> a=obj_new('idl_container')
IDL> a->[Tab]
JS> "Select a procedure-method name (class is IDL_Container):"
```

JS> The class is stored as a property on the "->" for rapid recovery if
JS> custom variable `idlwave-store-inquired-class' is set, and is even
JS> recallable from the input history (just look for the bold arrow).

JS> - Much improved class and structure tag completion and help. When

JS> completing class tags (e.g. self.[Tab]) or structure tags (when using
JS> the optional completion module idlw-complete-structtag), you can now
JS> right-click on an item to pop-up its source definition in the online
JS> Help window. Invoking context help on class tags (either with key
JS> command [M-S-?], or just with a shift-right-click on a self.tag
JS> expression) works too. This even works for inherited tags defined in
JS> superclasses (in which case the class of the tag will be appended in
JS> the completions buffer)! Never scroll to the bottom of your file, or
JS> hunt about for the superclass file(s) to find out what one of your
JS> class or structure tags means again.

JS> - Conditional breakpoints are now easily set using a simple
JS> prefix-argument to the breakpoint setting command (e.g. [C-u C-c C-d
JS> C-b], or, e.g., [C-u H-d], for those who set up an alternate prefix
JS> modifier). Conditional breakpoints allow any IDL expression (like "i
JS> gt 25") to be used as a break condition. You can still of course set
JS> one-time or "after" breakpoints using the same easy numeric prefixes
JS> (e.g. [C-4 C-c C-d C-d]).

JS> - Filename completion in the shell now works after batch '@' commands.

JS> - A custom variable 'idlwave-shell-show-commands' allows you to control
JS> which of the background commands sent to the shell are printed,
JS> including compiling, breakpoint, debug, and miscellaneous commands. A
JS> special choice ('everything') allows you to have a peek at all the
JS> traffic IDLWAVE is sending to the shell (for the insatiably curious).
JS> Pay no attention to the man behind the curtain.

JS> - When using stepping commands "Continue to Here" and "Continue to End
JS> of Block", all other breakpoints are temporarily disabled to ensure
JS> you reach the desired break location.

JS> - XEmacs binary "packages" of IDLWAVE are now available for download on
JS> each release.

JS> =====> Bug fixes and the like: <=====

JS> - Filename completion in the shell and buffer now works better, fixing
JS> the occasional appearance of cruft at the end of your filename.
JS> Also, you can now complete blank filenames (e.g. open,[Tab]).

JS> - Now compiles correctly with "downgrade" versions of help.

JS> - Multi-threading keywords are now correctly linked for both routine
JS> info and help.

JS> - Certain syntax errors weren't being parsed correctly: fixed.

JS> - Completion of tags in `self`, and in other structures (like info),
JS> when the optional completion module `idlw-complete-struct-tag` is
JS> used, now catches a few tags which were being missed.

JS> - Alignment on continued assignment statements now permits a more
JS> complex LHS (e.g. `a.x[4] = $ ...`).

JS> - An incorrect non-matching-end error when using multi-word end
JS> abbreviations like "\ei" ("endif else if") was quelled. Begin/end
JS> block showing now works more smoothly.

JS> - Fixed issue on completing system variable structure tags (like
JS> `!D.TABLE_SIZE`) when the optional `idlw-complete-struct-tag` completion
JS> module is also in use.

JS> - Completing procedures on multi-statement lines (with `&`) now works
JS> correctly.

JS> - Indenting continuation lines which contain the text "pro" or
JS> "function" somewhere in their argument list now works correctly.

JS> As always, all this and more is available for the taking at:

JS> <http://idlwave.org>

JS> Happy coding!

JS> JD
