

IDLWAVE Reference Card

IDLWAVE Version 6.0: www.idlwave.org

Editing IDL Programs

Indent the current line relative to context	TAB
Re-indent all lines in the current region	C-M-\
Re-indent all lines in the current routine	C-M-q
Re-indent all lines in the current statement	C-u TAB
Start a continuation line, splitting the current line at point	M-RET
Fill the current comment paragraph	M-q
Display calling sequence and keywords for the procedure or function call at point	C-c ?
Load context sensitive online help for nearby routine, keyword, etc	M-?
Complete a procedure name, function name or keyword in the buffer	M-TAB
Update IDLWAVE's knowledge about functions and procedures	C-c C-i
Visit the source code of a procedure/function in this buffer	C-c C-v
Visit the source code of a procedure/function insert a standard documentation header	C-u C-c C-v
Insert a standard documentation header	C-c C-h
Insert a new timestamp and history item in the documentation header	C-c RET

Motion

Beginning of subprogram	C-M-a
End of subprogram	C-M-e
Beginning of block (stay inside the block)	C-c {
End of block (stay inside the block)	C-c }
Forward block (on same level)	C-M-n
Backward block (on same level)	C-M-p
Down block (enters a block)	C-M-d
Backward up block (leaves a block)	C-M-u
Next Statement	C-c C-n

Code Templates

CASE statement template	C-c C-c
FOR loop template	C-c C-f
REPEAT loop template	C-c C-r
WHILE loop template	C-c C-w

Code Abbreviations

PROCEDURE template	\pr
FUNCTION template	\fu
CASE statement template	\c
FOR loop template	\f
REPEAT loop template	\r
WHILE loop template	\w
IF statement template	\i
IF-ELSE statement template	\elif
if n_elements() eq 0 then	\ine
if n_elements() ne 0 then	\inn

Code Abbreviations (continued)

arg_present()	\ap	begin	\b
byte()	\cb	complex()	\cc
double()	\cd	float()	\cf
long()	\cl	common	\co
string()	\cs	fix()	\cx
else	\e	endcase	\ec
endelse	\ee	endfor	\ef
endif else if	\ei	endif else	\el
endif	\en	endrep	\er
endswitch	\es	endwhile	\ew
goto,	\g	help,	\h
if keyword_set() then	\ik	if arg_present() then	\iap
keyword_set()	\k	n_elements()	\n
n_params()	\np	on_ioerror,	\oi
open,	\or	openu,	\ou
openw,	\ow	print,	\p
plot,	\pt	ptr_valid()	\pv
read,	\re	readf,	\rf
return	\rt	readu,	\ru
size()	\s	strcompress()	\sc
strlowcase()	\sl	strmid()	\sm
strlen()	\sn	strpos()	\sp
strtrim()	\sr	strput()	\st
strucase()	\su	then	\t
until	\u	widget_control,	\wc
widget_info()	\wi	writew,	\wu

IDLWAVE Shell, Debugging

Start IDL shell and/or switch to the shell	C-c C-s
Complete the name of procedure, function or keyword	TAB, M-TAB
Cycle back through IDL command history	↑, M-p
Cycle forward	↓, M-n
Save and compile the source file	C-c C-d C-c
Go to next syntax error	C-c C-d C-x
Switch to electric debug mode	C-c C-d C-v
Print the value of the expression near point	C-c C-d C-p
Print the help-information on the expression	C-c C-d ?
Previous input matching a regex	M-r
Next input matching a regex	M-s
Beginning of line; skip prompt	C-c C-a
Kill input to beginning of line	C-c C-u
Kill word before cursor	C-c C-w
Send ^C	C-c C-c
Send ^Z	C-c C-z
Send ^"	C-c C-\
Delete last batch of process output	C-c C-o
Show last batch of process output	C-c C-r
List input history	C-c C-l
Routine Info display	C-c ?
IDL online help on routine	M-?
Update routine info from buffers and shell	C-c C-i
Find the source file of a routine	C-c C-v
Find the source file of a routine in the currently visited file	C-c C-t
Compile a library routine	C-c =

Breakpoints and Stepping

Set breakpoint	C-c C-d C-b
Set breakpoint in module named here	C-c C-d C-i
Clear current breakpoint	C-c C-d C-d
Clear all breakpoints	C-c C-d C-a
Go to the previous breakpoint	C-c C-d [
Go to the next breakpoint	C-c C-d]
Disable/Enable current breakpoint	C-c C-d C-\
Set a breakpoint at the routine beginning	C-c C-d C-j
Step, into function calls	C-c C-d C-s
Step, over function calls(stepover)	C-c C-d C-n
Skip one statement	C-c C-d C-k
Continue to end of block	C-c C-d C-u
Continue to end of function	C-c C-d C-m
Continue past end of function	C-c C-d C-o
Continue to line at cursor position	C-c C-d C-h
Continue execution to next breakpoint	C-c C-d C-r
Show higher level in calling stack	C-c C-d C-↑
Show lower level in calling stack	C-c C-d C-↓

Electric Debug Mode

Clear all breakpoints	a
Set breakpoint, C-u b for a conditional break, C-n b for nth hit	b
Clear current breakpoint	d
Prompt for expression to print	e
Continue to the line at cursor position	h
Set breakpoint in module named here	i
Go to the previous breakpoint in the file	[
Go to the next breakpoint in the file]
Disable/Enable current breakpoint	\
Set breakpoint at beginning of enclosing routine	j
Skip one statement	k
Continue to end of function	m
Step, over function calls	n
Continue past end of function	o
Print expression near point or in region with C-u p	p
End the debugging session and return to the Shell's main level	q
Continue execution	r
Step, into function calls	s or SPACE
Print a calling-level traceback in the shell	t
Continue to end of block	u
Turn Electric Debug Mode off	v
Examine expression near point	x
Reset IDL	z
Show higher level in calling stack	+ or =
Show lower level in calling stack	- or _
Help on expression near point or in region with C-u ?	?
Show help on the commands available	C-?

Copyright © 2006 Maxim Neumann.
v0.2 for IDLWAVE version 6.0, May 2006

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all copies.