

QUICK REFERENCE

CONVEX CXdb

Order No. DSW-474

First Edition
June 1991



CONVEX

CONVEX COMPUTER CORPORATION

CONVEX

CXdb Quick Reference

Order No. DSW-474

First Edition

June 1991

Copyright 1991 CONVEX Computer Corporation.
All rights reserved.

CONVEX and the CONVEX logo "C" are registered
trademarks of CONVEX Computer Corporation.

UNIX is a trademark of AT&T Bell Laboratories.

Contents

Commands.....	1
Abbreviations.....	31
Default Aliases.....	35

Commands

add cmderr —Add viewports to cmderr.

add cmderr <viewport> [, ...]

add cmdlog —Add viewports to cmdlog.

add cmdlog <viewport> [, ...]

add cmdout —Add viewports to cmdout.

add cmdout <viewport> [, ...]

add default environment —Add or modify environment variables in the default environment.

add default environment <environment-variable> = <string> [, ...]

add default path —Add directories to the default search path.

add default path <directory-specifier> [, ...]

add environment —Add or modify environment variables in the process environment.

[<process-list>] **add environment** <environment-variable> = <string> [, ...]

add path —Add directories to the process search path.

[<process-list>] **add path** <directory-specifier> [, ...]

alias —Define an alias.

alias <alias-name> <string>

attach —Debug a running process.

[<process-list>] **attach** <process-id>

backtrace —Display the frames of the call stack.

[<process-list>] [<thread-list>] **backtrace** [<frame-count>]

bind —Define key bindings for Maryland Windows.

bind <function-name> <key-name>

break instruction —Set a breakpoint at an instruction.

[<process-list>] [<thread-list>] **break instruction** <language-expression> [{<event-handler>}][<debugger-variable>]

break line —Set a breakpoint at a source line.

[<process-list>] [<thread-list>] **break line** <line-specifier> [{<event-handler>}][<debugger-variable>].

break routine —Set a breakpoint at the beginning of a routine.

[<process-list>] [<thread-list>] **break routine** <language-expression> [{<event-handler>}][<debugger-variable>]

break source —Set a breakpoint at a source unit.

[<process-list>] [<thread-list>] **break source** <source-unit> [{<event-handler>}][<debugger-variable>]

cd —Change the console working directory.

cd <directory-specifier>

clear default environment —Remove all environment variables from the default environment.

clear default environment

clear default fixed sched —Disable fixed scheduling in the default settings.

clear default fixed sched

clear echo —Disable echoing of input from initialization files and command files.

clear echo

clear environment —Remove all environment variables from the process environment.

[<process-list>] **clear environment**

clear fixed sched —Disable fixed scheduling in the process settings.

[<process-list>] **clear fixed sched**

clear logging —Disable logging for cmdlog.

clear logging

clear noclobber —Disable the noclobber option for all viewports.

clear noclobber

clear seq —Clear the SEQ bit.

[<process-list>] [<thread-list>] **clear seq**

clear sqs —Clear the SQS bit.

[<process-list>] [<thread-list>] **clear sqs**

clear step —Reset the stepping granularity to the default setting.

[<process-list>] [<thread-list>] **clear step**

continue —Continue execution of the process.

[<process-list>] [<thread-list>] **continue** [<count>] [&]

copy —Copy a memory region.

[<process-list>] [<thread-list>] **copy** <source-address> [{ ..<ending-address> | :<byte-count> }] \; <destination-address>

core —Retrieve the image of a core file or a checkpoint file.

[<process-list>] **core** <file-name>

cxdb —Invoke CXdb from the shell.

cxdb [-a <process-id>] [-b] [-D <directory-specifier>[: ...]] [-f <file-name>] [-F] [-nw] [-nx] [-sw <geometry>] [-cw <geometry>] [-hw <geometry>] [-pw <geometry>] [-fw <geometry>] [-x <command-list>] [[-e <file-name>] [[-c <file-name>] [<x-toolkit-options>]

debug core —Debug a core file or a checkpoint file.

debug core <file-name> [<debugger-variable>]

debug exec —Debug an executable file.

debug exec <file-name> [<debugger-variable>]

debug proc —Debug a running process.

debug proc <process-id> [<debugger-variable>]

detach —Detach from a process.

[<process-list>] **detach**

disable event —Disable eventpoints.

disable event <event-specifier> [, ...]

disable eventtype —Disable all eventpoints of the specified type.

[<process-list>] **disable eventtype** <eventtype-specifier> [, ...]

disassemble —Display the disassembled code.

[<process-list>] [<thread-list>] **disassemble** [<starting-address> [{ ..<ending-address> | :<instruction-count> }]]

display file —Display the contents of a file.

[<process-list>] **display file** <file-name>

display routine —Display the source code for a routine.

[<process-list>] **display routine** <language-expression>

echo —Echo a character string.

echo[/n] <string> [...]

edit —Edit a file.

edit [*<file-name>*]

enable event —Enable eventpoints.

enable event *<event-specifier>* [, ...]

enable eventtype —Enable all eventpoints of the specified type.

[*<process-list>*] **enable eventtype** *<eventtype-specifier>* [, ...]

evaluate —Evaluate a language expression.

[*<process-list>*] [*<thread-list>*] **evaluate** *<language-expression>*

event exec —Set an eventpoint to watch for an `exec (2)` system call by the process.

[*<process-list>*] **event exec** [{*<event-handler>*}] [*<debugger-variable>*]

event join —Set an eventpoint to trap a thread joining.

[*<process-list>*] **event join** [{*<event-handler>*}] [*<debugger-variable>*]

event modify —Set an eventpoint to watch for a value change within an address range.

```
[<process-list>] [<thread-list>] event modify <starting-address> [{ ..<ending address> | :<byte-count> }] [ {<event-handler> } ]
[<debugger-variable>]
```

event reached instruction —Set an eventpoint at an instruction.

```
[<process-list>] [<thread-list>] event reached instruction <language-expression> [ {<event-handler> } ] [<debugger-variable>]
```

event reached line —Set an eventpoint at a source line.

```
[<process-list>] [<thread-list>] event reached line <line-specifier> [ {<event-handler> } ] [<debugger-variable>]
```

event reached routine —Set an eventpoint at the beginning of a routine.

```
[<process-list>] [<thread-list>] event reached routine <language-expression> [ {<event-handler> } ] [<debugger-variable>]
```

event reached source —Set an eventpoint at a source unit.

```
[<process-list>] [<thread-list>] event reached source <source-unit> [ {<event-handler> } ] [<debugger-variable>]
```

event relation —Set an eventpoint to watch for an expression to become true.

```
[<process-list>] [<thread-list>] event relation <language-expression> [ {<event-handler> } ] [<debugger-variable>]
```

event signal —Set an eventpoint to catch a signal.

[<process-list>] **event signal** <signal-specifier> [{<event-handler>}] [<debugger-variable>]

event spawn —Set an eventpoint to trap the spawning of a thread.

[<process-list>] **event spawn** [{<event-handler>}] [<debugger-variable>]

examine —Display a region of memory.

[<process-list>] [<thread-list>] **examine** [/<memory-unit> <format> <fpmode>]] <starting-address>
[[..<ending-address> | :<unit-count>]]

executable —Load an executable file for debugging.

[<process-list>] **executable** <file-name>

fill —Fill a region of memory with the value of an expression.

[<process-list>] [<thread-list>] **fill** [/<memory-unit> <starting-address> [[..<ending-address> | :<unit-count>]] \;
<language-expression>

find memory backward —Find a byte pattern within a memory region.

[<process-list>] [<thread-list>] **find memory backward** <byte-pattern> <lowest-address> { ..<highest-address> | :<byte-count>}

find memory forward —Find a byte pattern within a memory region.

[<process-list>] [<thread-list>] **find memory forward** <byte-pattern> <starting-address> { ..<ending-address> | :<byte-count>}

find window backward —Find a character string in a source window.

find window backward <string> [{ <window-number> | <debugger-variable> }]

find window forward —Find a character string in a source window.

find window forward <string> [{ <window-number> | <debugger-variable> }]

finish —Finish executing (step out of) the specified source unit.

[<process-list>] [<thread-list>] **finish** [<granularity>] [&]

frame —Change the current stack frame.

[<process-list>] [<thread-list>] **frame** <frame-specifier>

goto address —Branch to the specified address.

[<process-list>] [<thread-list>] **goto address** <language-expression>

goto line —Branch to the specified source line.

[<process-list>] [<thread-list>] **goto line** <line-specifier>

goto source —Branch to the specified source unit.

[<process-list>] [<thread-list>] **goto source** <source-unit>

help —Invoke the CXdb help system.

help [<string>]

if —Establish conditional execution of CXdb commands.

if (<relational-expression>) <command-set> [**else** <command-set>]

info alias —Display aliases.

info alias [<regular-expression>]

info args —Display arguments of the current routine.

[<process-list>] [<thread-list>] **info args**

info bind —Display key bindings for Maryland Windows.

info bind [<key-name>]

info break —Display all existing breakpoints.

[<process-list>] [<thread-list>] **info break**

info cregisters —Display the communication registers.

[<process-list>] **info cregisters**

info cxdb —Display the status of the current CXdb session.

info cxdb

info default environment —Display all default environment variables.

info default environment [<regular-expression>]

Info environment —Display all process environment variables.
[<process-list>] **info environment** [<regular-expression>]

Info errno —Display the error message received by the process.
[<process-list>] [<thread-list>] **info errno**

Info event —Display the specified eventpoints.
info event [<event-specifier>] [, ...]

Info eventtype —Display all eventpoints of the specified type.
[<process-list>] **info eventtype** <eventtype-specifier> [, ...]

Info expression —Display the characteristics of the specified language expression.
[<process-list>] [<thread-list>] **info expression** <language-expression>

Info formatting —Display the settings for memory display formats.
[<process-list>] **info formatting**

info frame —Display a stack frame.

[*<process-list>*] [*<thread-list>*] **info frame** [*<frame-specifier>*]

info frame at —Display the stack frame at the specified address.

[*<process-list>*] [*<thread-list>*] **info frame at** *<language-expression>*

info history —Display the CXdb command history.

info history [*<command-count>*]

info line —Display the source units on a specified line.

[*<process-list>*] **info line** *<line-specifier>*

info locals —Display the local variables of the current routine.

[*<process-list>*] [*<thread-list>*] **info locals**

info macro —Display macros.

info macro [*<regular-expression>*]

info objectmap —Display the object map.

[<process-list>] **info objectmap**

info process —Display the status of the current process.

[<process-list>] **info process**

info psw —Display the processor status word.

[<process-list>] [<thread-list>] **info psw**

info registers —Display the scalar and address registers.

[<process-list>] [<thread-list>] **info registers**

info scope —Display the current scope path.

[<process-list>] [<thread-list>] **info scope**

info signal —Display the settings of the signal actions.

[<process-list>] **info signal** [<signal-specifier>] [, ...]

Info sourceunit —Display the specified source unit.
[<process-list>] [<thread-list>] **info sourceunit** <source-unit>

Info stack —Display the process stack.
[<process-list>] [<thread-list>] **info stack**

Info symbols —Display program symbols.
[<process-list>] **info symbols** [<regular-expression>]

Info threads —Display threads of the current process.
[<process-list>] [<thread-list>] **info threads**

Info trace —Display all tracepoints.
[<process-list>] [<thread-list>] **info trace**

Info type —Display type definitions.
[<process-list>] [<thread>] **info type** [<regular-expression>]

info vregisters —Display the vector registers.

[<process-list>] [<thread-list>] **info vregisters**

info watch —Display all watchpoints.

[<process-list>] [<thread-list>] **info watch**

kill process —Terminate the process.

[<process-list>] **kill process**

macro —Define a macro.

macro <name> [(<parameter>[:<default>] [, ...])] <string>

next —Step to the next source unit, ignoring subroutine calls.

[<process-list>] [<thread-list>] **next** [<granularity>] [<count>] [&]

next instruction —Step to the next instruction, ignoring subroutine calls.

[<process-list>] [<thread-list>] **next instruction** [<count>] [&]

next over —Step from the current source unit of specified granularity to the next source unit of default granularity, ignoring subroutine calls.

[<process-list>] [<thread-list>] **next over** [<granularity>] [<count>] [&]

print —Evaluate a language expression and print the result.

[<process-list>] [<thread-list>] **print**[/[n]<format><fpmode>] <language-expression>

pwd —Display the name of the console working directory.

pwd

quit —Exit from CXdb.

quit

recall —Re-execute a previous command.

recall [?]<string>

remove alias —Delete an alias.

remove alias <alias-name>

remove cmderr —Delete viewports from cmderr.

remove cmderr <viewport> [, ...]

remove cmdlog —Delete viewports from cmdlog.

remove cmdlog <viewport> [, ...]

remove cmdout —Delete viewports from cmdout.

remove cmdout <viewport> [, ...]

remove default environment —Delete environment variables from the default environment.

remove default environment <environment-variable> [, ...]

remove default path —Delete directories from the default search path.

remove default path <directory-specifier> [, ...]

remove environment —Delete environment variables from the process environment.

[<process-list>] **remove environment** <environment-variable> [, ...]

remove event —Delete the specified eventpoints.

remove event *<event-specifier>* [, ...]

remove eventtype —Delete all eventpoints of the specified type.

[*<process-list>*] **remove eventtype** *<eventtype-specifier>* [, ...]

remove macro —Delete a macro.

remove macro *<name>*

remove path —Delete directories from the process search path.

[*<process-list>*] **remove path** *<directory-specifier>* [, ...]

remove variable —Delete a debugger variable.

remove variable *<debugger-variable>*

rerun —Start execution of the process, using the previous argument list.

[*<process-list>*] **rerun** [*&*]

resume —Continue execution of the process from within an eventpoint handler.

resume

return —Return to the calling routine.

[<process-list>] [<thread-list>] **return** <language-expression>

run —Start execution of the process.

[<process-list>] **run** [<string>] [&]

set cmderr —Clear and redefine the viewport list for cmderr.

set cmderr <viewport> [, ...]

set cmdlog —Clear and redefine the viewport list for cmdlog.

set cmdlog <viewport> [, ...]

set cmdout —Clear and redefine the viewport list for cmdout.

set cmdout <viewport> [, ...]

set default environment —Clear and redefine the environment variables for the default environment.

```
set default environment <environment-variable> = <string> [, ...]
```

set default fixed sched —Enable fixed scheduling in the default settings.

```
set default fixed sched
```

set default format —Set the default formats for displaying memory.

```
set default format <memory-unit> <format>
```

set default fpmode —Set the default floating point mode for new processes.

```
set default fpmode { ieee | native | dual }
```

set default handler —Set the default handler for eventpoints.

```
set default handler {<event-handler>}
```

set default memory —Set the default unit size for displaying memory.

```
set default memory <memory-unit>
```

set default path —Set the default search path.

set default path <directory-specifier> [, ...]

set default pshell —Set the default process shell.

set default pshell {sh | csh}

set default step —Set the default stepping granularity.

set default step <granularity>

set directory —Set the process working directory.

[<process-list>] **set directory** <directory-specifier>

set echo —Enable echoing of input from command files.

set echo

set environment —Clear and redefine the environment variables for the process environment.

[<process-list>] **set environment** <environment-variable>=<string> [, ...]

set evalopts fpmode —Set the floating point mode for evaluating expressions.

`set evalopts fpmode { ieee | native | dual }`

set evalopts iprecision —Set the size of integer constants for CXdb.

`set evalopts iprecision { 4 | 8 }`

set evalopts rprecision —Set the size of real numbers for CXdb.

`set evalopts rprecision { 4 | 8 }`

set fixed sched —Enable fixed scheduling in the process settings.

`[<process-list>] set fixed sched`

set format —Set the formats for displaying memory.

`[<process-list>] [<thread-list>] set format <memory-unit> <format>`

set fpmode —Set the floating point mode for the process.

`[<process-list>] set fpmode { ieee | native }`

set handler —Set the handler for a specified eventpoint.

set handler *<event-specifier>* [, ...] {*<event-handler>*}

set ignore —Set an ignore count for an eventpoint.

set ignore *<ignore-count>* *<event-specifier>* [, ...]

set logging —Enable logging for cmdlog.

set logging

set memory —Set the unit size for displaying memory.

[*<process-list>*] [*<thread-list>*] **set memory** *<memory-unit>*

set noclobber —Enable the noclobber option for all viewports.

set noclobber

set path —Set the search path for the process.

[*<process-list>*] **set path** *<directory-specifier>* [, ...]

set printopts maxarray —Set the maximum number of array elements to print.

set printopts maxarray <number-of-elements>

set printopts nopadding —Disable padding with leading zeros when printing.

set printopts nopadding

set printopts padding —Enable padding with leading zeros when printing.

set printopts padding

set printopts precision —Set the precision used to print floating point numbers.

set printopts precision <width>.<precision>

set pshell —Set the process shell.

[<process-list>] **set pshell** {sh | csh}

set seq —Set the SEQ bit.

[<process-list>] [<thread-list>] **set seq**

set shell —Set the type of shell invoked from within CXdb.

set shell (**sh** | **cs**h | **tc**sh | **k**sh | **COVUE**)

set signal —Set the actions for the specified signal.

[<process-list>] **set signal** <signal-specifier> [(**stop** | **nostop**)] [(**pass** | **nopass**)] [(**print** | **no**print)]

set sqs —Set the SQS bit.

[<process-list>] [<thread-list>] **set sqs**

set step —Set the stepping granularity.

[<process-list>] [<thread-list>] **set step** <granularity>

set typehandler —Define the default handler for all eventpoints of the specified type.

set typehandler <eventtype-specifier> [, ...] {<event-handler>}

shell —Invoke a shell.

shell [/<shell-specifier>] [<shell-commands>]

signal process —Send a signal to the process.

[<process-list>] **signal process** <signal-specifier> [&]

signal thread —Send a signal to a specific thread of the process.

[<process-list>] <thread> **signal thread** <signal-specifier> [&]

source —Execute a CXdb command file.

source <file-name>

step —Step to the next source unit.

[<process-list>] [<thread-list>] **step** [<granularity>] [<count>] [&]

step instruction —Step to the next instruction.

[<process-list>] [<thread-list>] **step instruction** [<count>] [&]

step over —Step from the current source unit of specified granularity to the next source unit of default granularity.

[<process-list>] [<thread-list>] **step over** [<granularity>] [<count>] [&]

stop —Stop execution of the process.

[<process-list>] **stop**

trace instruction —Set a tracepoint at an instruction.

[<process-list>] [<thread-list>] **trace instruction** <language-expression> [{<event-handler>}] [<debugger-variable>]

trace line —Set a tracepoint at a source line.

[<process-list>] [<thread-list>] **trace line** <line-specifier> [{<event-handler>}] [<debugger-variable>]

trace routine —Set a tracepoint at the beginning of a routine.

[<process-list>] [<thread-list>] **trace routine** <language-expression> [{<event-handler>}] [<debugger-variable>]

trace source —Set a tracepoint at a source unit.

[<process-list>] [<thread-list>] **trace source** <source-unit> [{<event-handler>}] [<debugger-variable>]

watch —Set a watchpoint to monitor an address range.

[<process-list>] [<thread-list>] **watch** <starting-address> [{ ..<ending address> | :<byte-count>}] [{<event-handler>}]
[<debugger-variable>]

Abbreviations

ad cmde	add cmderr
ad cmdl	add cmdlog
ad cmdo	add cmdout
ad d e	add default environment
ad d p	add default path
ad e	add environment
ad p	add path
al	alias
at	attach
ba	backtrace
bin	bind
bre i	break instruction
bre l	break line
bre r	break routine
bre s	break source
cd	cd
cl d e	clear default environment
cl d f s	clear default fixed sched
cl ec	clear echo
cl en	clear environment
cl f s	clear fixed sched
cl l	clear logging
cl n	clear noclobber
cl se	clear seq
cl sq	clear sqs
cl st	clear step
con	continue
cop	copy
cor	core
cxdb	cxdb
deb c	debug core
deb e	debug exec
deb p	debug proc
det	detach
disab event	disable event
disab eventt	disable eventtype
disas	disassemble
disp f	display file
disp r	display routine
ec	echo
ed	edit
ena event	enable event
ena eventt	enable eventtype
eva	evaluate
eve e	event exec
eve j	event join
eve m	event modify
eve rea i	event reached instruction
eve rea l	event reached line
eve rea r	event reached routine
eve rea s	event reached source

eve rel	event relation
eve si	event signal
eve sp	event spawn
exa	examine
exe	executable
fil	fill
find m b	find memory backward
find m f	find memory forward
find w b	find window backward
find w f	find window forward
fini	finish
fr	frame
g a	goto address
g l	goto line
g s	goto source
h	help
if	if
in al	info alias
in ar	info args
in bi	info bind
in br	info break
in cr	info cregisters
in cx	info cxdb
in de	info default environment
in en	info environment
in er	info errno
in event	info event
in eventt	info eventtype
in ex	info expression
in fo	info formatting
in fr	info frame
in fr a	info frame at
in h	info history
in li	info line
in lo	info locals
in m	info macro
in o	info objectmap
in pr	info process
in ps	info psw
in r	info registers
in sc	info scope
in si	info signal
in so	info sourceunit
in st	info stack
in sy	info symbols
in th	info threads
in tr	info trace
in ty	info type
in v	info vregisters
in w	info watch
ki p	kill process
m	macro
n	next
n i	next instruction

n o	next over
pr	print
pw	pwd
q	quit
rec	recall
rem a	remove alias
rem cmde	remove cmderr
rem cmdl	remove cmdlog
rem cmdo	remove cmdout
rem d e	remove default environment
rem d p	remove default path
rem en	remove environment
rem event	remove event
rem eventt	remove eventtype
rem m	remove macro
rem p	remove path
rem v	remove variable
rer	rerun
res	resume
ret	return
ru	run
se cmde	set cmderr
se cmdl	set cmdlog
se cmdo	set cmdout
se de e	set default environment
se de fi s	set default fixed sched
se de fo	set default format
se de fp	set default fpmode
se de h	set default handler
se de m	set default memory
se de pa	set default path
se de ps	set default pshell
se de s	set default step
se di	set directory
se ec	set echo
se en	set environment
se ev f	set evalopts fpmode
se ev i	set evalopts iprecision
se ev r	set evalopts rprecision
se fi s	set fixed sched
se fo	set format
se fp	set fpmode
se h	set handler
se i	set ignore
se l	set logging
se m	set memory
se n	set noclobber
se pa	set path
se pr m	set printopts maxarray
se pr n	set printopts nopadding
se pr pa	set printopts padding
se pr pr	set printopts precision
se ps	set pshell
se se	set seq

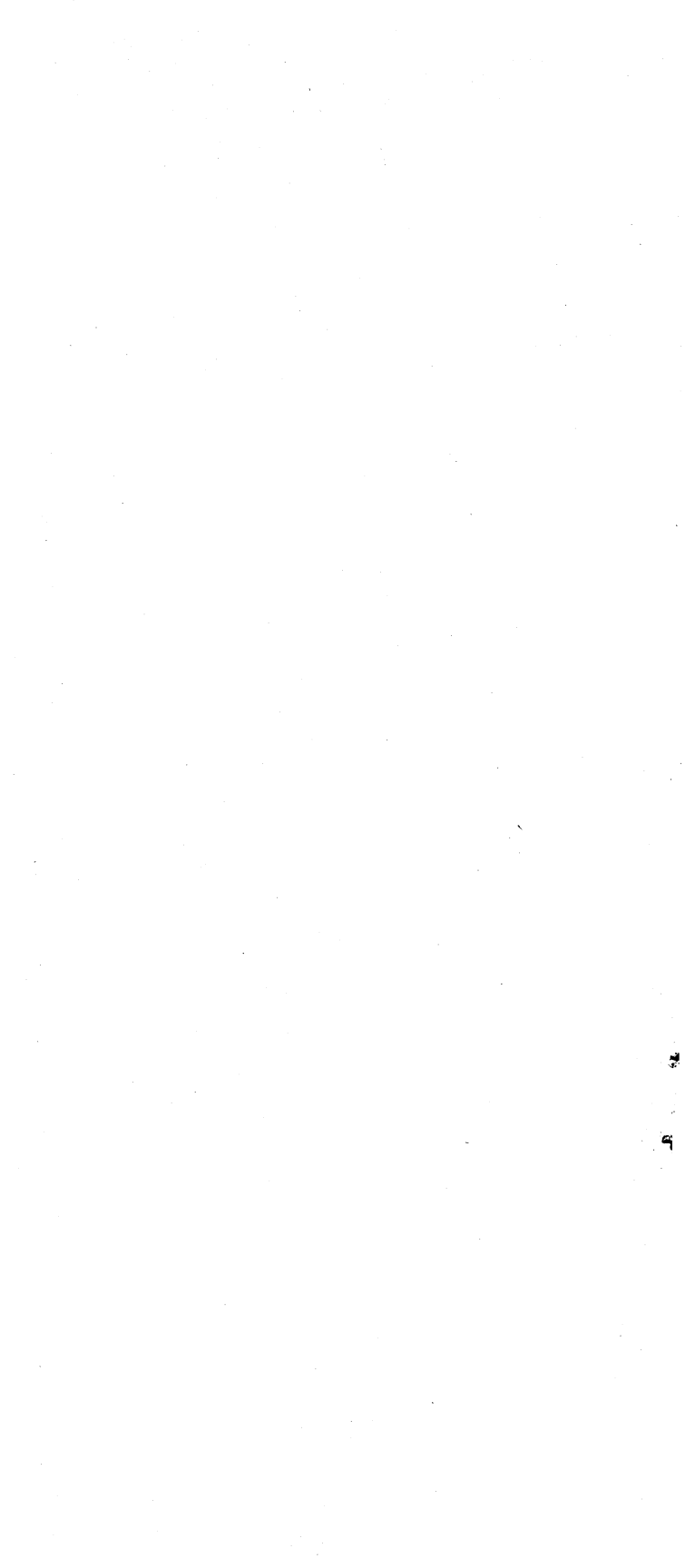
se sh	set shell
se si	set signal
se sq	set sqs
se st	set step
se t	set typehandler
sh	shell
sig p	signal process
sig t	signal thread
sou	source
ste	step
ste i	step instruction
ste o	step over
sto	stop
t i	trace instruction
t l	trace line
t r	trace routine
t s	trace source
w	watch

Default Aliases

```
denv+  add default environment
dp+    add default path
env+   add environment
p+     add path
bt     backtrace
bi     break instruction
bl     break line
br     break routine
bs     break source
cfs    clear fixed sched
c      continue
dbgc   debug core
dbg    debug exec
dbgp   debug proc
dis    disable event
en     enable event
x      examine
fmb    find memory backward
fmf    find memory forward
fwb    find window backward
fwf    find window forward
f      frame
down   frame -1
up     frame +1
?      help
i      info
args   info args
b?     info break
env?   info environment
e?     info event
et?    info eventtype
describe, whatis info expression
locals info locals
p?     info process
where  info scope
globals, symbols info symbols
t?     info trace
k      kill process
ni, nexti next instruction
no     next over
p      print
!      recall
denv-  remove default environment
dp-    remove default path
env-   remove environment
e-     remove event
et-    remove eventtype
p-     remove path
rr     rerun
r      run
denv=  set default environment
```

dp= set default path
env= set environment
sfs set fixed sched
p= set path
. source
s step
si, stepi step instruction
so step over
ti trace instruction
tl trace line
tr trace routine
ts trace source





CONVEX Computer Corporation
Richardson, Texas USA

CONVEX CXdb
Quick Reference
Part No. 710-015630-000