Debugging

Track program execution.

Debugging commands are essential to find programming errors. They help with inspecting programs and data, to pause a program, to find out why a program takes too much time, and much more. Logo can pause because of several reasons:

- The PAUSE command has been executed
- A runtime error has occured
- The PAUSE button has been clicked
- A STEPPED element has triggered a pause

BACKTRACE (BT)

Prints the list of nested procedure calls.

Syntax

```
BACKTRACE
(BACKTRACE number)
```

Description

In a debugging session, BACKTRACE prints the list of currently executing procedures. This is the same list as displayed in the Stack panel. If a number is used as input to BACKTRACE, the command outputs the stack trace belonging to the background engine whose ID matches the input. The number 0 always refers to the main Logo engine. See LAUNCH, HALT and :LOGOENGINE for more information.

If this command is used when the debugger is not active, it does nothing.

BPCLEAR

Deletes all breakpoints.

Syntax

```
BPCLEAR
```

Description

BPCLEAR deletes all breakpoints with all procedures.

CONTINUE (CO)

Ends a pause.

Syntax

```
CONTINUE
```

Description

CONTINUE ends a pause. A Logo program can pause because is it is stepped (see STEP), the PAUSE command has been executed, or if Logo has hit a runtime error.

If this command is used when the debugger is not active, it does nothing.

PAUSE

Pauses a procedure.

Syntax

```
PAUSE
```

Description

PAUSE temporarily halts the execution of a procedure. PAUSE makes it possible to check variables or change the environment during the execution of a procedure. To resume execution of the procedure, press the Continue button or type CO or CONTINUE. To return to toplevel, press the STOP button or type TOPLEVEL.

If this command is used outside of a procedure, it does nothing.

It is more convenient to set a breakpoint in the editor by clicking the line number of the line where you want Logo to pause.

Example

```
TO SQUAREDRAW
    FORWARD 60
    RT 90
    PAUSE
    REPEAT 3 [FD 60 RT 90]
END

SQUAREDRAW
```

STEP

Turns on stepping for the given elements.

Syntax

```
STEP procedurename
STEP [procedurename1 ...]
STEP contents-list
```

Description

STEP turns the stepping on for the given elements. If the input is a list or a word, stepping is turned on for the given procedures. If the input is a structured contents list, stepping is turned on for the given procedures, names and properties.

When a procedure is being stepped, Logo pauses at the beginning of that procedure and enters debug mode. This is the

same as starting the procedure definition with a PAUSE command.

When a name is being stepped, Logo pauses whenever the value of the name is changed, or the name is erased, and enters debug mode. Note that the name must be a global name; local names cannot be stepped.

When a property list is being stepped, Logo pauses whenever any value of the property is changed, or a property is erased, and enters debug mode.

STEPPED? (STEPPEDP)

Outputs TRUE if the element described by its input is stepped.

Syntax

```
STEPPED? name or list
```

Description

STEPPED? outputs TRUE if the element described by its input is stepped. If the input is a list or a structured contents list, STEPPED? outputs the stepped state of the first non-empty element of the list.

STEPPED

Outputs a structured contents list of all stepped elements.

Syntax

```
STEPPED
```

Description

STEPPED outputs a structured contents list of all stepped, non-empty elements. For an explanation of stepping see STEP.

TRACE

Turns on tracing of the given elements.

Syntax

```
TRACE procedurename
TRACE [procedurename1 ...]
TRACE contents-list
```

Description

TRACE turns the tracing on for the given elements. If the input is a list or a word, tracing is turned on for the given procedures. If the input is a structured contents list, tracing is turned on for the given procedures, names, and properties.

When a procedure is being traced, Logo prints a message to the Output panel every time the procedure is entered or left. When entering the procedure, the call is printed along with the procedure arguments. When the procedure leaves, the

message contains the value that the procedure outputs, if any.

When a name is being traced, Logo prints a message to the Output panel whenever the value for the name is changed along with the new value, or when the value is erased. Also, Logo prints a message if a local variable hides a global variable (or a local variable in a calling procedure) that has the same name.

When a property list is being stepped, Logo prints a message to the Output panel whenever any value of the property is changed, or a property is erased, and enters debug mode.

Note that the global variable `:TRACE` takes precedence over the settings of this command. Also, TRACE does not display each line of a procedure as it is being executed, but `:TRACE` does.

TRACED? (TRACEDP)

Outputs TRUE if the element described by its input is traced.

Syntax

```
TRACED? name or list
```

Description

TRACED? outputs TRUE if the element described by its input is traced. If the input is a list or a structured contents list, TRACED? outputs the traced state of the first non-empty element of the list.

TRACED

Outputs a structured contents list of all traced elements.

Syntax

```
TRACED
```

Description

TRACED outputs a structured contents list of all traced, non-empty elements. For an explanation of tracing see TRACE.

UNSTEP

Turns off stepping of the given elements.

Syntax

```
UNSTEP procedurename
UNSTEP [procedurename1 ...]
UNSTEP contents-list
```

Description

UNSTEP turns the stepping off for the given elements. If the input is a list or a word, stepping is turned off for the given

procedures. If the input is a structured contents list, stepping is turned off for the given procedures, names and properties. For an explanation of stepping, see STEP.

UNTRACE

Turns off tracing of the given elements.

Syntax

```
UNTRACE procedurename
UNTRACE [procedurename1 ...]
UNTRACE contents-list
```

Description

UNTRACE turns the tracing off for the given elements. If the input is a list or a word, tracing is turned off for the given procedures. If the input is a structured contents list, tracing is turned off for the given procedures, names, and properties. For an explanation of tracing, see TRACE.