# Contents

## 1 lackey classes
1.1 lackey.Region class ......................................................... 3
1.2 lackey.Pattern class .......................................................... 12
1.3 lackey.Match class ............................................................. 13
1.4 lackey.Screen class ............................................................ 13
1.5 lackey.Location class ........................................................ 14
1.6 lackey.Mouse class ............................................................. 15
1.7 lackey.Keyboard class ........................................................ 17
1.8 lackey.App class ............................................................... 17
1.9 lackey.Button class ........................................................... 18
1.10 lackey.Key class ............................................................... 18
1.11 lackey.KeyModifier class .................................................. 20
1.12 lackey.PlatformManagerWindows class ................................ 20
1.13 lackey.Debug class ........................................................... 20
1.14 lackey.Settings class ........................................................ 22
1.15 lackey.FindFailed class ..................................................... 22
1.16 Sikuli Convenience Functions ............................................ 23

## 2 PlatformManager API
2.1 Introduction ........................................................................... 25
2.2 Properties ............................................................................ 25
2.3 Methods ............................................................................... 25
1.1 lackey.Region class

class lackey.Region(*args):
    Bases: object

    ABORT = 'ABORT'
    BB = 211
    CREATE_X_DIRECTION_LEFT = 0
    CREATE_X_DIRECTION_RIGHT = 1
    CREATE_Y_DIRECTION_BOTTOM = 1
    CREATE_Y_DIRECTION_TOP = 0
    EAST = 220
    EAST_MID = 310
    LL = 210
    MID_BIG = 'MID_HALF'
    MID_HORIZONTAL = 'MID_HORZ'
    MID_THIRD = 311
    MID_VERTICAL = 'MID_VERT'
    NORTH = 202
    NORTH_EAST = 302
    NORTH_MID = 301
    NORTH_WEST = 300
    PROMPT = 'PROMPT'
RETRY = 'RETRY'
RR = 201
SKIP = 'SKIP'
SOUTH = 212
SOUTH_EAST = 322
SOUTH_MID = 321
SOUTH_WEST = 320
TT = 200
WEST = 221
WEST_MID = 312
above (expand=None)
    Returns a new Region above the current region with a height of expand pixels.
    Does not include the current region. If range is omitted, it reaches to the top of the screen. The new region
    has the same width and x-position as the current region.
aboveAt (offset=0)
    Returns point in the center of the region’s top side (offset to the top by negative offset)
add(l, r, t, b)
asOffset ()
    Returns bottom right corner as offset from top left corner
atMouse ()
below (expand=None)
    Returns a new Region below the current region with a height of expand pixels.
    Does not include the current region. If range is omitted, it reaches to the bottom of the screen. The new
    region has the same width and x-position as the current region.
bottomAt (offset=0)
    Returns point in the center of the region’s bottom side (offset to the bottom by offset)
    click (target=None, modifiers="")
        Moves the cursor to the target location and clicks the default mouse button.
    clipRegionToScreen ()
        Returns the part of the region that is visible on a screen
        If the region equals to all visible screens, returns Screen(-1). If the region is visible on multiple screens,
        returns the screen with the smallest ID. Returns None if the region is outside the screen.
    compare (image)
        Compares the region to the specified image
    contains (point_or_region)
        Checks if point_or_region is within this region
    containsMouse ()
    copyTo (screen)
classmethod create (*args)
**debugPreview** *(title='Debug')*

Displays the region in a preview window.

If the region is a Match, circles the target area. If the region is larger than half the primary screen in either
dimension, scales it down to half size.

**delayType**

**doubleClick** *(target=None, modifiers="")*

Moves the cursor to the target location and double-clicks the default mouse button.

**drag** *(dragFrom=None)*

Moves the cursor to the target location and clicks the mouse in preparation to drag a screen element.

**dragDrop** *(target, target2=None, modifiers="")*

Performs a dragDrop operation.

Holds down the mouse button on dragFrom, moves the mouse to dragTo, and releases the mouse button.

**dropAt** *(dragTo=None, delay=None)*

Completes a dragDrop operation.

Moves the cursor to the target location, waits delay seconds, and releases the mouse button.

**exists** *(pattern, seconds=None)*

Searches for an image pattern in the given region.

Returns Match if pattern exists, None otherwise (does not throw exception). Sikuli supports OCR search
with a text parameter. This does not (yet).

**find** *(pattern)*

Searches for an image pattern in the given region.

Throws FindFailed exception if the image could not be found. Sikuli supports OCR search with a text
parameter. This does not (yet).

**findAll** *(pattern)*

Searches for an image pattern in the given region.

Returns Match object if pattern exists, empty array otherwise (does not throw exception). Sikuli
supports OCR search with a text parameter. This does not (yet).

**findAllByRow** *(target)*

Returns an array of rows in the region (defined by the raster), each row containing all matches in that row
for the target pattern.

**findAllBycolumn** *(target)*

Returns an array of columns in the region (defined by the raster), each column containing all matches in
that column for the target pattern.

**findAllText** *(text)*

OCR function.

**findBest** *(pattern)*

Returns the best match in the region (instead of the first match).

**findText** *(text, timeout=None)*

OCR function.
get \((part)\)
   Returns a section of the region as a new region

   Accepts partitioning constants, e.g. Region.NORTH, Region.NORTH_WEST, etc. Also accepts an int
   200-999: * First digit: Raster \((n\) rows by \(n\) columns) * Second digit: Row index (if equal to raster, gets the
   whole row) * Third digit: Column index (if equal to raster, gets the whole column)

   Region.get(522) will use a raster of 5 rows and 5 columns and return the cell in the middle.
   Region.get(525) will use a raster of 5 rows and 5 columns and return the row in the middle.

getAutoWaitTimeout()
   Returns the time to wait for an image to appear on the screen

getBitmap()
   Captures screen area of this region, at least the part that is on the screen

   Returns image as numpy array

getBottomLeft()
   Return the Location of the bottom left corner of this region

getBottomRight()
   Return the Location of the bottom right corner of this region

ggetCell\((row, column)\)
   Returns the specified cell (if a raster is set for the region)

ggetCell\(W()\)

gGetCols()

gEvent\(\text{name}\)
   Returns the named event.

   Removes it from the internal queue.

gEvents()
   Returns a list of all events that have occurred.

   Empties the internal queue.

gFindFailedResponse()
   Returns the current default response to a FindFailed exception

gH()
   Get the height of the region

gLastMatch()
   Returns the last successful Match returned by find(), exists(), etc.
**getLastMatches**
Returns the last successful set of Match objects returned by `findAll()`.

**getScreenImage**
Gets the last image taken on this region’s screen.

**getObserveScanRate**
Gets the number of times per second the observe loop should run.

**getRepeatWaitTime**
Gets the wait time before repeating a search.

**getRow** *(row, numberRows=None)*
Returns the specified row of the region (if the raster is set)
If numberRows is provided, uses that instead of the raster.

**getRowH**

**getRows**

**getScreen**
Return an instance of the Screen object this region is inside.
Checks the top left corner of this region (if it touches multiple screens) is inside. Returns None if the region isn’t positioned in any screen.

**getTarget**
By default, a region’s target is its center.

**getThrowException**
Returns True if an exception will be thrown for FindFailed operations, False otherwise.

**getTime**
Returns the elapsed time in milliseconds to find the last match.

**getTopLeft**
Return the Location of the top left corner of this region.

**getTopRight**
Return the Location of the top right corner of this region.

**getTuple**
Returns the shape of the region as (x, y, w, h).

**getW**
Get the width of the region.

**getWaitScanRate**
Get the current scan rate.

**getX**
Get the x-coordinate of the upper left-hand corner.

**getY**
Get the y-coordinate of the upper left-hand corner.

**grow** *(width, height=None)*
Expands the region by width on both sides and height on the top and bottom.
If only one value is provided, expands the region by that amount on all sides. Equivalent to `nearby()`.

**hasEvents**
Check whether any events have been caught for this region.

---

### 1.1. lackey.Region class

---
hasObserver()
  Check whether at least one event is registered for this region.
  
The observer may or may not be running.
highlight(*args)
  Highlights the region with a colored frame. Accepts the following parameters:
  highlight(toEnable, [seconds], [color])
  • toEnable (boolean): Enables or disables the overlay
  • seconds (number): Seconds to show overlay
  • color (string): Hex code (“#XXXXXX”) or color name (“black”)
hover(target=None)
  Moves the cursor to the target location
inside()
  Returns the same object. Included for Sikuli compatibility.
intersection()
  Returns a new region that contains the overlapping portion of this region
  and the specified region (may be None)
isChanged(min_changed_pixels, screen_state)
  Returns true if at least min_changed_pixels are different between screen_state
  and the current state.
isObserving()
  Check whether an observer is running for this region
isRasterValid()
isRegionValid()
  Returns false if the whole region is not even partially inside any screen, otherwise true
keyDown(keys)
  Concatenate multiple keys to press them all down.
keyUp(keys)
  Concatenate multiple keys to up them all.
left(expand=None)
  Returns a new Region left of the current region with a width of expand pixels.
  Does not include the current region. If range is omitted, it reaches to the left
  border of the screen. The new region has the same height and y-position as the current region.
leftAt(offset=0)
  Returns point in the center of the region’s left side (offset to the left
  by negative offset)
morphTo(region)
  Change shape of this region to match the given Region object
mouseDown(button=<Mock id='140108365020944'>)
  Low-level mouse actions.
mouseMove(PSRML=None, dy=0)
  Low-level mouse actions
mouseUp(button=<Mock id='140108365020944'>)
  Low-level mouse actions

Chapter 1. lackey classes
moveTo (location)
Change the upper left-hand corner to a new Location

Doesn’t change width or height

nearby (expand=50)
Returns a new Region that includes the nearby neighbourhood of the the current region.
The new region is defined by extending the current region’s dimensions all directions by range number of pixels. The center of the new region remains the same.

observe (seconds=None)
Begins the observer loop (synchronously).
Loops for seconds or until this region’s stopObserver() method is called. If seconds is None, the observer loop cycles until stopped. If this method is called while the observer loop is already running, it returns False.

Returns True if the observer could be started, False otherwise.

observeInBackground (seconds=None)
As Region.observe(), but runs in a background process, allowing the rest of your script to continue.

Note that the subprocess operates on copies of the usual objects, not the original Region object itself for example. If your event handler needs to share data with your main process, check out the documentation for the multiprocessing module to set up shared memory.

offset (location, dy=0)
Returns a new Region offset from this one by location

Width and height remain the same

onAppear (pattern, handler=None)
Registers an event to call handler when pattern appears in this region.
The handler function should take one parameter, an ObserveEvent object (see below). This event is ignored in the future unless the handler calls the repeat() method on the provided ObserveEvent object.

Returns the event’s ID as a string.

onChange (min_changed_pixels=None, handler=None)
Registers an event to call handler when at least min_changed_pixels change in this region.
(Default for min_changed_pixels is set in Settings.ObserveMinChangedPixels)
The handler function should take one parameter, an ObserveEvent object (see below). This event is ignored in the future unless the handler calls the repeat() method on the provided ObserveEvent object.

Returns the event’s ID as a string.

onVanish (pattern, handler=None)
Registers an event to call handler when pattern disappears from this region.
The handler function should take one parameter, an ObserveEvent object (see below). This event is ignored in the future unless the handler calls the repeat() method on the provided ObserveEvent object.

Returns the event’s ID as a string.

paste (*args)
Usage: paste([PSMRL], text)
If a pattern is specified, the pattern is clicked first. Doesn’t support text paths. text is pasted as is using the OS paste shortcut (Ctrl+V for Windows/Linux, Cmd+V for OS X). Note that paste() does NOT use special formatting like type().
Lackey Documentation, Release 0.7.3

resetScreens()
   Synonym for resetMonitors

right(expand=None)
   Returns a new Region right of the current region with a width of expand pixels.
   Does not include the current region. If range is omitted, it reaches to the right border of the screen. The
   new region has the same height and y-position as the current region.

rightAt(offset=0)
   Returns point in the center of the region’s right side (offset to the right by offset)

rightClick(target=None, modifiers=””)
   Moves the cursor to the target location and clicks the right mouse button.

saveLastScreenImage()
   Saves the last image taken on this region’s screen to a temporary file

saveScreenCapture(path=None, name=None)
   Saves the region’s bitmap

setActive(name)
   Activates an inactive event type.

setAutoWaitTimeout(seconds)
   Specify the time to wait for an image to appear on the screen

setBottomLeft(loc)
   Move this region so its bottom left corner is on loc

setBottomRight(loc)
   Move this region so its bottom right corner is on loc

setCenter(loc)
   Move this region so it is centered on loc

setCols(columns)
   Sets the number of columns in the raster (if rows have not been initialized, set to 1 as well)

setFindFailedHandler(handler)
   Set a handler to receive FindFailed events (instead of triggering an exception).

setFindFailedResponse(response)
   Set the response to a FindFailed exception in this region.
   Can be ABORT, SKIP, PROMPT, or RETRY.

setH(h)
   Set the height of the region

setImageMissingHandler(handler)
   Set a handler to receive ImageMissing events (instead of triggering an exception).

setInactive(name)
   The specified event is ignored until reactivated or until the observer restarts.

setLocation(location)
   Change the upper left-hand corner to a new Location
   Doesn’t change width or height

setObserveScanRate(scan_rate)
   Set the number of times per second the observe loop should run
setROI(*args)
Set Region of Interest (same as Region.setRect())

setRaster(rows, columns)
Sets the raster for the region, allowing sections to be indexed by row/column

setRect(*args)
Sets the rect of the region. Accepts the following arguments:
setRect(rect_tuple) setRect(x, y, w, h) setRect(rect_region)

setRepeatWaitTime(wait_time)
Sets the wait time before repeating a search

setRows(rows)
Sets the number of rows in the raster (if columns have not been initialized, set to 1 as well)

setSize(w, h)
Sets the new size of the region

setThrowException(setting)
Defines whether an exception should be thrown for FindFailed operations.
setting should be True or False.

setTopLeft(loc)
Move this region so its top left corner is on loc

setTopRight(loc)
Move this region so its top right corner is on loc

setWidth(w)
Set the width of the region

setWaitScanRate(seconds=None)
Set this Region's scan rate
A find op should repeat the search for the given Visual rate times per second until found or the maximum
waiting time is reached.

setX(x)
Set the x-coordinate of the upper left-hand corner

setY(y)
Set the y-coordinate of the upper left-hand corner

showScreens()
Synonym for showMonitors

stopObserver()
Stops this region’s observer loop.
If this is running in a subprocess, the subprocess will end automatically.

text()
OCR method. Todo.

type(*args)
Usage: type([PSMRL], text, [modifiers])
If a pattern is specified, the pattern is clicked first. Doesn’t support text paths.
Special keys can be entered with the key name between brackets, as “[SPACE]”, or as Key.SPACE.
union()

Returns a new region that contains both this region and the specified region

wait(pattern, seconds=None)

Searches for an image pattern in the given region, given a specified timeout period

Functionally identical to find(). If a number is passed instead of a pattern, just waits the specified number of seconds. Sikuli supports OCR search with a text parameter. This does not (yet).

waitVanish(pattern, seconds=None)

Waits until the specified pattern is not visible on screen.

If seconds pass and the pattern is still visible, raises FindFailed exception. Sikuli supports OCR search with a text parameter. This does not (yet).

wheel(*args)

Clicks the wheel the specified number of ticks. Use the following parameters:

wheel([PSRML], direction, steps, [stepDelay])

write(text)

Has fancy special options. Not implemented yet.

1.2 lackey.Pattern class

class lackey.Pattern(target=None)

Bases: object

Defines a pattern based on a bitmap, similarity, and target offset

deprecatedPreview(title='Debug')

Loads and displays the image at Pattern.path

exact()

Returns a new Pattern with a similarity threshold of 1.0

getFilename()

Returns the path to this Pattern’s bitmap

getImage()

getSimilar()

Returns the current minimum similarity

getTargetOffset()

Returns the target offset as a Location(dx, dy)

isImagePattern()

isValid()

setFilename(filename)

Set the filename of the pattern’s image (and load it)

setImage(img)

similar(similarity)

Returns a new Pattern with the specified similarity threshold

targetOffset(dx, dy)

Returns a new Pattern with the given target offset
1.3 lackey.Match class

class lackey.Match(score, target, rect)
    Bases: lackey.RegionMatching.Region

    Extended Region object with additional data on click target, match score

    getScore()
        Returns confidence score of the match

    getTarget()
        Returns the location of the match click target (center by default, but may be offset)

1.4 lackey.Screen class

class lackey.Screen(screenId=None)
    Bases: lackey.RegionMatching.Region

    Individual screen objects can be created for each monitor in a multi-monitor system.

    Screens are indexed according to the system order. 0 is the primary monitor (display 1), 1 is the next monitor, etc.

    Lackey also makes it possible to search all screens as a single “virtual screen,” arranged according to the system’s settings. Screen(-1) returns this virtual screen. Note that the larger your search region is, the slower your search will be, so it’s best practice to adjust your region to the particular area of the screen where you know your target will be.

    Note that Sikuli is inconsistent in identifying screens. In Windows, Sikuli identifies the first hardware monitor as Screen(0) rather than the actual primary monitor. However, on OS X it follows the latter convention. We’ve opted to make Screen(0) the actual primary monitor (wherever the Start Menu/System Menu Bar is) across the board.

    capture(*args)
        Captures the region as an image

    captureForHighlight(*args)
        Captures the region as an image

    closePrompt()
        Not yet implemented

    doPrompt(message, obs)
        Not yet implemented

    getBounds()
        Returns bounds of screen as (x, y, w, h)

    getCurrentID()
        Returns screen ID

    getID()
        Returns screen ID

    getLastScreenImageFromScreen()
        Returns the last captured image from this screen

    classmethod getNumberScreens()
        Get the number of screens in a multi-monitor environment at the time the script is running
1.5 lackey.Location class

class lackey.Location(x, y)
    Bases: object
    Basic 2D point object

    above(dy)
    Get a new location dy pixels vertically above the current location.

    below(dy)
    Get a new location dy pixels vertically below the current location.

    click()

    copyTo(screen)
    Creates a new point with the same offset on the target screen as this point has on the current screen

    doubleClick()

    getColor()
getMonitor()
Returns an instance of the Screen object this Location is inside.

Returns the primary screen if the Location isn’t positioned in any screen.

getOffset(loc)
Returns the offset between the given point and this point

getScreen()
Returns an instance of the Screen object this Location is inside.

Returns None if the Location isn’t positioned in any screen.

gETuple()
Returns coordinates as a tuple (for some PlatformManager methods)

getX()
Returns the X-component of the location

getY()
Returns the Y-component of the location

grow(*args)
Creates a region around the given point Valid arguments:
  • grow(wh) - Creates a region centered on this point with a width and height of wh.
  • grow(w, h) - Creates a region centered on this point with a width of w and height of h.
  • grow(Region.CREATE_X_DIRECTION, Region.CREATE_Y_DIRECTION, w, h) - Creates a region with this point as one corner, expanding in the specified direction

hover()

left(dx)
Get a new location dx pixels horizontally to the left of the current location.

move(x, y)
Set the location of this object to the specified coordinates.

moveTo(x, y)
Set the location of this object to the specified coordinates.

offset(dx, dy)
Get a new location which is dx and dy pixels away horizontally and vertically from the current location.

right(dx)
Get a new location dx pixels horizontally to the right of the current location.

setLocation(x, y)
Set the location of this object to the specified coordinates.

translate(dx, dy)

1.6 lackey.Mouse class

class lackey.Mouse
    Bases: object

    Mid-level mouse routines.
    LEFT
MIDDLE

RIGHT

WHEEL_DOWN = 0

WHEEL_UP = 1

at()

Gets Location of cursor

buttonDown (button=<Mock id='140108365054416'>)

Holds down the specified mouse button.

Use Mouse.LEFT, Mouse.MIDDLE, Mouse.RIGHT

buttonUp (button=<Mock id='140108365054608'>)

Releases the specified mouse button.

Use Mouse.LEFT, Mouse.MIDDLE, Mouse.RIGHT

click (loc=None, button=<Mock id='140108365054288'>)

Clicks the specified mouse button.

If loc is set, move the mouse to that Location first.

Use button constants Mouse.LEFT, Mouse.MIDDLE, Mouse.RIGHT

down (button=<Mock id='140108365054416'>)

Holds down the specified mouse button.

Use Mouse.LEFT, Mouse.MIDDLE, Mouse.RIGHT

getPos()

Gets Location of cursor

hasMoved()

Checks if mouse was moved since last mouse action

move (loc, yoff=None)

Moves cursor to specified location. Accepts the following arguments:

• move (loc) - Move cursor to Location
• move (xoff, yoff) - Move cursor to offset from current location

moveSpeed (location, seconds=0.3)

Moves cursor to specified Location over seconds.

If seconds is 0, moves the cursor immediately. Used for smooth somewhat-human-like motion.

up (button=<Mock id='140108365054608'>)

Releases the specified mouse button.

Use Mouse.LEFT, Mouse.MIDDLE, Mouse.RIGHT

wheel (direction, steps)

Clicks the wheel the specified number of steps in the given direction.

Use Mouse.WHEEL_DOWN, Mouse.WHEEL_UP
1.7 lackey.Keyboard class

class lackey.Keyboard
    Bases: object
    Mid-level keyboard routines. Interfaces with PlatformManager
    
    keyDown (keys)
    Accepts a string of keys (including special keys wrapped in brackets or provided by the Key or KeyModifier classes). Holds down all of them.
    
    keyUp (keys)
    Accepts a string of keys (including special keys wrapped in brackets or provided by the Key or KeyModifier classes). Releases any that are held down.
    
    type (text, delay=0.1)
    Translates a string into a series of keystrokes.
    Respects Sikuli special codes, like “{ENTER}”.

1.8 lackey.App class

class lackey.App (identifier=None)
    Bases: object
    Allows apps to be selected by title, PID, or by starting an application directly. Can address individual windows tied to an app.
    
    For more information, see Sikuli’s App documentation.
    
    classmethod close (appName)
    Closes the process associated with the specified app.
    As a class method, accessible as App.class(appName). As an instance method, accessible as App(appName).close().
    
    classmethod focus (appName)
    Searches for exact text, case insensitive, anywhere in the window title.
    Brings the matching window to the foreground.
    As a class method, accessible as App.focus(appName). As an instance method, accessible as App(appName).focus().
    
    classmethod focusedWindow ()
    Returns a Region corresponding to whatever window is in the foreground
    
    classmethod getClipboard ()
    Gets the contents of the clipboard (as classmethod)
    
    getName ()
    Returns the short name of the app as shown in the process list
    
    getPID ()
    Returns the PID for the associated app (or -1, if no app is associated or the app is not running)
    
    getWindow ()
    Returns the title of the main window of the currently open app.
    Returns an empty string if no match could be found.
hasWindow()
    Returns True if the process has a window associated, False otherwise

isRunning(waitTime=0)
    If PID isn’t set yet, checks if there is a window with the specified title.

isValid()

classmethod open(executable)
    Runs the specified command and returns an App linked to the generated PID.

    As a class method, accessible as App.open(executable_path). As an instance method, accessible as App(executable_path).open().

classmethod pause(waitTime)

classmethod setClipboard(contents)
    Sets the contents of the clipboard (as classmethod)

setUsing(params)

waitForWindow(seconds=5)

window(windowNum=0)
    Returns the region corresponding to the specified window of the app.

    Defaults to the first window found for the corresponding PID.

1.9 lackey.Button class

class lackey.Button

    CENTER = 1
    LEFT = 0
    RIGHT = 2

1.10 lackey.Key class

class lackey.Key

    Key codes for InputEmulation.Keyboard object.

    Can be entered directly or concatenated with an existing string, e.g. type(Key.TAB)

    ADD = '{ADD}'
    ALT = '{ALT}'
    BACKSPACE = '{BACKSPACE}'
    CAPS_LOCK = '{CAPS_LOCK}'
    CMD = '{CMD}'
    CTRL = '{CTRL}'
    DELETE = '{DELETE}'
    DIVIDE = '{DIVIDE}'
Lackey Documentation, Release 0.7.3

1.10. lackey.Key class

```
DOWN = '{DOWN}'
END = '{END}'
ENTER = '{ENTER}'
ESC = '{ESC}'
F1 = '{F1}'
F10 = '{F10}'
F11 = '{F11}'
F12 = '{F12}'
F13 = '{F13}'
F14 = '{F14}'
F15 = '{F15}'
F16 = '{F16}'
F2 = '{F2}'
F3 = '{F3}'
F4 = '{F4}'
F5 = '{F5}'
F6 = '{F6}'
F7 = '{F7}'
F8 = '{F8}'
F9 = '{F9}'
HOME = '{HOME}'
INSERT = '{INSERT}'
LEFT = '{LEFT}'
META = '{META}'
MINUS = '{MINUS}'
MULTIPLY = '{MULTIPLY}'
NUM0 = '{NUM0}'
NUM1 = '{NUM1}'
NUM2 = '{NUM2}'
NUM3 = '{NUM3}'
NUM4 = '{NUM4}'
NUM5 = '{NUM5}'
NUM6 = '{NUM6}'
NUM7 = '{NUM7}'
NUM8 = '{NUM8}'
NUM9 = '{NUM9}'
```
NUM_LOCK = '{NUM_LOCK}'
PAGE_DOWN = '{PAGE_DOWN}'
PAGE_UP = '{PAGE_UP}'
PAUSE = '{PAUSE}'
PRINTSCREEN = '{PRINTSCREEN}'
RIGHT = '{RIGHT}'
SCROLL_LOCK = '{SCROLL_LOCK}'
SEPARATOR = '{SEPARATOR}'
SHIFT = '{SHIFT}'
SPACE = '{SPACE}'
TAB = '{TAB}'
UP = '{UP}'
WIN = '{WIN}'

# 1.11 lackey.KeyModifier class

class lackey.KeyModifier
    Can be used with type() to modify another key, e.g. type(Key.DELETE, Key.CTRL+Key.ALT)
    ALT = '{ALT}'
    CMD = '{CMD}'
    CTRL = '{CTRL}'
    META = '{META}'
    SHIFT = '{SHIFT}'
    WIN = '{WIN}'

# 1.12 lackey.PlatformManagerWindows class

lackey.PlatformManagerWindows
    alias of lackey.PlatformManagerWindows

# 1.13 lackey.Debug class

class lackey.DebugMaster
    Bases: object
    Used to create the global Debug object
    error(message)
        Records an Error-level log message
        Uses the log path defined by Debug.setUserLogFile(). If no log file is defined, sends to STDOUT
**history** *(message)*
Records an Action-level log message

Uses the log path defined by `Debug.setUserLogFile()`. If no log file is defined, sends to STDOUT

**info** *(message)*
Records an Info-level log message

Uses the log path defined by `Debug.setUserLogFile()`. If no log file is defined, sends to STDOUT

**log** *(level, message)*
Records a Debug-level log message Uses the log path defined by `Debug.setUserLogFile()`. If no log file is defined, sends to STDOUT

**off** ()
Turns off all debugging messages

**on** *(level)*
Turns on all debugging messages up to the specified level

0 = None; 1 = User;

**setLogFile** *(filepath)*
Defines the file to which output log messages should be sent.

Set to `None` to print to STDOUT instead.

**setLogger** *(logger_obj)*
Sets log handler to `logger_obj`

**setLoggerAction** *(mthd)*
Sends action messages to `logger.[mthd]()` for handling

**setLoggerAll** *(mthd)*
Sends all messages to `logger.[mthd]()` for handling

**setLoggerDebug** *(mthd)*
Sends debug messages to `logger.[mthd]()` for handling

**setLoggerError** *(mthd)*
Sends error messages to `logger.[mthd]()` for handling

**setLoggerInfo** *(mthd)*
Sends info messages to `logger.[mthd]()` for handling

**setLoggerNoPrefix** *(logger_obj)*
Sets log handler to `logger_obj`

**setLoggerUser** *(mthd)*
Sends user messages to `logger.[mthd]()` for handling

**user** *(message)*
Creates a user log (if user logging is turned on)

Uses the log path defined by `Debug.setUserLogFile()`. If no log file is defined, sends to STDOUT

Note: Does **not** use Java string formatting like Sikuli. Format your message with Python `basestring.format()` instead.
1.14 lackey.Settings class

class lackey.SettingsMaster
    Bases: object
    Global settings that Lackey refers to by default
    ActionLogs = True
    BundlePath = '/home/docs/checkouts/readthedocs.org/user_builds/lackey/envs/latest/bin'
    ClickDelay = 0.0
    DebugLogs = False
    DelayBeforeDrag = 0.3
    DelayBeforeDrop = 0.3
    DelayBeforeMouseDown = 0.3
    ErrorLogs = False
    ImagePaths = []
    InfoLogs = True
    LogTime = False
    MinSimilarity = 0.7
    MoveMouseDelay = 0.3
    ObserveMinChangedPixels = 50
    ObserveScanRate = 3
    OcrDataPath = None
    PopupLocation = None
    ShowActions = False
    SlowMotionDelay = 3
    TypeDelay = 0.0
    UserLogPrefix = 'user'
    UserLogTime = True
    UserLogs = True
    WaitScanRate = 3
    getSikuliVersion()

1.15 lackey.FindFailed class

class lackey.FindFailed(event)
    Bases: exceptions.Exception
    Exception: Unable to find the searched item
1.16 Sikuli Convenience Functions

lackey.**sleep**(seconds)
Convenience function. Pauses script for seconds.

lackey.**exit**(value)
Convenience function. Exits with code value.

lackey.**setShowActions**(value)
Convenience function. Sets “show actions” setting (True or False)

lackey.**getBundlePath**()
Convenience function. Returns the path of the *.sikuli bundle.

lackey.**getBundleFolder**()
Convenience function. Same as getBundlePath() plus the OS default path separator.

lackey.**setBundlePath**(path)
Convenience function. Changes the path of the *.sikuli bundle.

lackey.**getImagePath**()
Convenience function. Returns a list of paths to search for images.

lackey.**addImagePath**(new_path)
Convenience function. Adds a path to the list of paths to search for images.

Can be a URL (but must be accessible).

lackey.**addHTTPImagePath**(new_path)
Convenience function. Same as addImagePath().

lackey.**getParentPath**()
Convenience function. Returns the parent folder of the *.sikuli bundle.

lackey.**getParentFolder**()
Convenience function. Same as getParentPath() plus the OS default path separator.

lackey.**makePath**(args)
Convenience function. Returns a path from a series of path components.

Same as os.path.join.

lackey.**makeFolder**(args)
Convenience function. Same as makePath() plus the OS default path separator.

lackey.**unzip**(from_file, to_folder)
Convenience function.

Extracts files from the zip file fromFile into the folder toFolder.

lackey.**popat**(args)
Convenience function. Sets the popup location (currently not used).

lackey.**popup**(text, title='Lackey Info')
Creates an info dialog with the specified text.

lackey.**popError**(text, title='Lackey Error')
Creates an error dialog with the specified text.

lackey.**popAsk**(text, title='Lackey Decision')
Creates a yes-no dialog with the specified text.
lackey.input (msg="", default="", title='Lackey Input', hidden=False)
    Creates an input dialog with the specified message and default text.
    If hidden, creates a password dialog instead. Returns the entered value.

lackey.inputText (message="", title='Lackey Input', lines=9, width=20, text="")
    Creates a textarea dialog with the specified message and default text.
    Returns the entered value.

lackey.select (message="", title='Lackey Input', options=None, default=None)
    Creates a dropdown selection dialog with the specified message and options
    default must be one of the options.
    Returns the selected value.

lackey.popFile (title='Lackey Open File')
    Creates a file selection dialog with the specified message and options.
    Returns the selected file.
2.1 Introduction

OS-specific functionality is abstracted into a PlatformManager object for ease of porting Lackey to other desktop platforms. Any object which obeys the interface definition provided herein can be used as a PlatformManager. New PlatformManagers should be added to the conditional statements in RegionMatching.py.

2.2 Properties

PlatformManager has no public properties. Accessing private properties of a specific PlatformManager from outside the module is bad practice and is strongly discouraged.

2.3 Methods

- **Clipboard Functions**
  - `osCopy ()`:
    * Triggers the OS “copy” keyboard shortcut (usually CTRL+C or CMD+C)
  - `osPaste ()`:
    * Triggers the OS “paste” keyboard shortcut (usually CTRL+V or CMD+V)

- **Window Functions**
  - `getWindowByTitle (wildcard, windowNum=0)`:
    * Returns the native window handle (as a Python object) for a window that matches the specified regex. If multiple windows with the regex exist, return the `windowNum`-th of them (starting from 0).
  - `getWindowByPID (pid, windowNum=0)`:
* Returns the native window handle (as a Python object) for a window that is owned by the specified process. If multiple windows exist for the specified process, return the \texttt{windowNum}-th of them (starting from 0).

– \texttt{getForegroundWindow} ():
  * Returns the native window handle (as a Python object) for the window currently in the foreground.

– \texttt{getWindowPID} (\texttt{handle}):
  * Return the PID that owns the given native window handle (provided by a function like \texttt{getWindowByTitle()}). Returns \texttt{-1} if the process no longer exists.

– \texttt{getWindowRect} (\texttt{handle}):
  * Return a rect \((x1,y1,x2,y2)\) for the area of the window with the given native handle.

– \texttt{getWindowTitle} (\texttt{handle}):
  * Return the title of the window with the given native handle.

– \texttt{focusWindow} (\texttt{handle}):
  * Bring the specified window to the front and give it focus.

• \textbf{Screen Functions}

– \texttt{getBitmapFromRect} (\texttt{x, y, w, h}):
  * Returns a numpy array of the specified area of the screen. If the area goes outside the virtual screen rect, truncate the area at the edge. If part of the area is outside of a visible screen (but inside the virtual screen rect), set it to black.

– \texttt{getScreenBounds} (\texttt{screen}):
  * Returns the screen size of the specified monitor (0 being the primary monitor, 1+ being additional monitors; -1 to get the bounds of the virtual screen)

– \texttt{getScreenDetails} ():
  * Returns a list of the screens attached to the system. Each screen object has one property, "rect", which is a tuple containing the rect \((x, y, w, h)\) of the screen’s area relative to the main monitor.

– \texttt{isPointVisible} (\texttt{x, y}):
  * Checks if a point is visible (on any monitor).

• \textbf{Highlighting Functions}

– \texttt{highlight} (\texttt{rect, seconds}):
  * Draws a red rectangle around the region defined by \texttt{rect} which disappears after \texttt{seconds}.

• \textbf{Process Functions}

– \texttt{isPIDValid} (\texttt{pid}):
  * Returns \texttt{True} if there is a running process associated with the PID, \texttt{False} otherwise

– \texttt{killProcess} (\texttt{pid}):
  * Tries to terminate the process associated with the specified PID.

– \texttt{getProcessName} (\texttt{pid}):
  * Returns the module path (executable name) for the process associated with the specified PID.
### Supported Key Codes ###

- **Special Keys**
  - `{BACKSPACE}`
  - `{TAB}`
  - `{CLEAR}`
  - `{ENTER}`
  - `{SHIFT}`
  - `{CTRL}`
  - `{ALT}`
  - `{PAUSE}`
  - `{CAPS_LOCK}`
  - `{ESCAPE}`
  - `{SPACE}`
  - `{PAGE_UP}`
  - `{PAGE_DOWN}`
  - `{END}`
  - `{HOME}`
  - `{LEFT}`
  - `{UP}`
  - `{RIGHT}`
  - `{DOWN}`
  - `{SELECT}`
  - `{PRINT}`
  - `{PRINTSCREEN}`
  - `{INSERT}`
  - `{DELETE}`
  - `{WIN}`
  - `{NUM_0}`
  - `{NUM_1}`
  - `{NUM_2}`
  - `{NUM_3}`
  - `{NUM_4}`
  - `{NUM_5}`
  - `{NUM_6}`
  - `{NUM_7}`
  - `{NUM_8}`
28 Chapter 2. PlatformManager API

- `{NUM_9}`
- `{F1}`
- `{F2}`
- `{F3}`
- `{F4}`
- `{F5}`
- `{F6}`
- `{F7}`
- `{F8}`
- `{F9}`
- `{F10}`
- `{F11}`
- `{F12}`
- `{F13}`
- `{F14}`
- `{F15}`
- `{F16}`
- `{NUM_LOCK}`
- `{SCROLL_LOCK}`
- `{[]}
- `{[]}
- `{+}
- `{@}
- `{^}
- `{}%
- `{~}
- `{()}
- `{())
- `{}}
- `{)}
Index

A
ABORT (lackey.Region attribute), 3
above() (lackey.Location method), 14
above() (lackey.Region method), 4
aboveAt() (lackey.Region method), 4
ActionLogs (lackey.SettingsMaster attribute), 22
ADD (lackey.Key attribute), 18
add() (lackey.Region method), 4
addHTTPImagePath() (in module lackey), 23
addImagePath() (in module lackey), 23
ALT (lackey.Key attribute), 18
ALT (lackey.KeyModifier attribute), 20
App (class in lackey), 17
asOffset() (lackey.Region method), 4
at() (lackey.Mouse method), 16
atMouse() (lackey.Region method), 4

B
BACKSPACE (lackey.Key attribute), 18
BB (lackey.Region attribute), 3
below() (lackey.Location method), 14
drag() (lackey.Region method), 4
bottomAt() (lackey.Region method), 4
BundlePath (lackey.SettingsMaster attribute), 22
Button (class in lackey), 18
download() (lackey.Mouse method), 16
downloadUp() (lackey.Mouse method), 16

C
CAPS_LOCK (lackey.Key attribute), 18
capture() (lackey.Screen method), 13
captureForHighlight() (lackey.Screen method), 13
CENTER (lackey.Button attribute), 18
drag() (lackey.Location method), 14
drag() (lackey.Mouse method), 16
drag() (lackey.Region method), 4
dragDelay (lackey.SettingsMaster attribute), 22
dropRegionToScreen() (lackey.Region method), 4
close() (lackey.App class method), 17
closePrompt() (lackey.Screen method), 13
CMD (lackey.Key attribute), 18
CMD (lackey.KeyModifier attribute), 20
copyTo() (lackey.Region method), 4
copyTo() (lackey.Region method), 4
cREATE_X_DIRECTION_LEFT (lackey.Region attribute), 3
cREATE_X_DIRECTION_RIGHT (lackey.Region attribute), 3
cREATE_Y_DIRECTION_BOTTOM (lackey.Region attribute), 3
cREATE_Y_DIRECTION_TOP (lackey.Region attribute), 3
CTRL (lackey.Key attribute), 18
CTRL (lackey.KeyModifier attribute), 20

D
DebugLogs (lackey.SettingsMaster attribute), 22
DebugMaster (class in lackey), 20
drag() (lackey.Location method), 14
drag() (lackey.Region method), 4
dragDelay (lackey.SettingsMaster attribute), 22
dragDelay (lackey.SettingsMaster attribute), 22
dragDelay (lackey.SettingsMaster attribute), 22
drawType() (lackey.Region method), 5
DELETE (lackey.Key attribute), 18
DIVIDE (lackey.Key attribute), 18
doPrompt() (lackey.Screen method), 13
dragClick() (lackey.Location method), 14
dragClick() (lackey.Region method), 5
DOWN (lackey.Key attribute), 18
down() (lackey.Mouse method), 16
drag() (lackey.Region method), 5
dragDrop() (lackey.Region method), 5
dropAt() (lackey.Region method), 5
E
EAST (lackey.Region attribute), 3
EAST_MID (lackey.Region attribute), 3
END (lackey.Key attribute), 19
ENTER (lackey.Key attribute), 19
error() (lackey.DebugMaster method), 20
ErrorLogs (lackey.SettingsMaster attribute), 22
ESC (lackey.Key attribute), 19
exact() (lackey.Pattern method), 12
exists() (lackey.Region method), 5
exit() (in module lackey), 23

F
F1 (lackey.Key attribute), 19
F10 (lackey.Key attribute), 19
F11 (lackey.Key attribute), 19
F12 (lackey.Key attribute), 19
F13 (lackey.Key attribute), 19
F14 (lackey.Key attribute), 19
F15 (lackey.Key attribute), 19
F16 (lackey.Key attribute), 19
F2 (lackey.Key attribute), 19
F3 (lackey.Key attribute), 19
F4 (lackey.Key attribute), 19
F5 (lackey.Key attribute), 19
F6 (lackey.Key attribute), 19
F7 (lackey.Key attribute), 19
F8 (lackey.Key attribute), 19
F9 (lackey.Key attribute), 19
find() (lackey.Region method), 5
findAll() (lackey.Region method), 5
findAllByColumn() (lackey.Region method), 5
findAllByRow() (lackey.Region method), 5
findAllText() (lackey.Region method), 5
findBest() (lackey.Region method), 5
FindFailed (class in lackey), 22
findText() (lackey.Region method), 5
focus() (lackey.App class method), 17
focusedWindow() (lackey.App class method), 17

G
get() (lackey.Region method), 5
getAutoWaitTimeout() (lackey.Region method), 6
getBitmap() (lackey.Region method), 6
getBottomLeft() (lackey.Region method), 6
getBottomRight() (lackey.Region method), 6
getBounds() (lackey.Screen method), 13
getBundleFolder() (in module lackey), 23
getBundlePath() (in module lackey), 23
getCell() (lackey.Region method), 6
getCenter() (lackey.Region method), 6
getClipboard() (lackey.Region method), 6
getClipboard() (lackey.App class method), 17
getCol() (lackey.Region method), 6
color() (lackey.Location method), 14
cols() (lackey.Region method), 6
colW() (lackey.Region method), 6
currentID() (lackey.Screen method), 13
event() (lackey.Region method), 6
events() (lackey.Region method), 6
filename() (lackey.Pattern method), 12
getFindFailedResponse() (lackey.Region method), 6
H (lackey.Region method), 6
ID() (lackey.Screen method), 13
getID() (lackey.Screen method), 13
getImage() (lackey.Pattern method), 12
getImagePath() (in module lackey), 23
getLastMatch() (lackey.Region method), 6
getLastMatches() (lackey.Region method), 6
getLastScreenImage() (lackey.Region method), 7
getLastScreenImageFromScreen() (lackey.Screen method), 13
getMonitor() (lackey.Location method), 14
getName() (lackey.App method), 17
getNumberOfScreens() (lackey.Screen class method), 13
getObserveScanRate() (lackey.Region method), 7
getOffset() (lackey.Location method), 15
getParentFolder() (in module lackey), 23
getParentPath() (in module lackey), 23
getPID() (lackey.App method), 17
getPost() (lackey.Mouse method), 16
getPrimaryID() (lackey.Screen class method), 13
getPrimaryScreen() (lackey.Screen class method), 14
geretRepeatWaitTime() (lackey.Region method), 7
getRow() (lackey.Region method), 7
getRowHi() (lackey.Region method), 7
getRows() (lackey.Region method), 7
getScore() (lackey.Match method), 7
getScreen() (lackey.Location method), 15
getScreen() (lackey.Region method), 7
getSikuliVersion() (lackey.SettingsMaster method), 22
getSimilar() (lackey.Pattern method), 12
getTarget() (lackey.Match method), 13
getTarget() (lackey.Region method), 7
geretTargetOffset() (lackey.Pattern method), 12
getThrowException() (lackey.Region method), 7
time() (lackey.Region method), 7
topLeft() (lackey.Region method), 7
topRight() (lackey.Region method), 7
tuple() (lackey.Location method), 15
tuple() (lackey.Region method), 7
W() (lackey.Region method), 7
getWaitScanRate() (lackey.Region method), 7
getWindow() (lackey.App method), 17
getX() (lackey.Location method), 15
getX() (lackey.Region method), 7
ty() (lackey.Location method), 15
ty() (lackey.Region method), 7
Index

M
makeFolder() (in module lackey), 23
makePath() (in module lackey), 23
Match (class in lackey), 13
META (lackey.Key attribute), 19
META (lackey.KeyModifier attribute), 20
MID_BIG (lackey.Region attribute), 3
MID_HORIZONTAL (lackey.Region attribute), 3
MID_THIRD (lackey.Region attribute), 3
MID_VERTICAL (lackey.Region attribute), 3
MIDDLE (lackey.Mouse attribute), 15
MinSimilarity (lackey.SettingsMaster attribute), 22
MINUS (lackey.Key attribute), 19
morphTo() (lackey.Region method), 8
Mouse (class in lackey), 15
mouseDown() (lackey.Region method), 8
mouseMove() (lackey.Region method), 8
mouseUp() (lackey.Region method), 8
move() (lackey.Location method), 15
move() (lackey.Mouse method), 16
MouseMoveDelay (lackey.SettingsMaster attribute), 22
moveSpeed() (lackey.Mouse method), 16
moveTo() (lackey.Location method), 15
moveTo() (lackey.Region method), 8
MULTIPLY (lackey.Key attribute), 19

N
nearby() (lackey.Region method), 9
newLocation() (lackey.Screen method), 14
newRegion() (lackey.Screen method), 14
NORTH (lackey.Region attribute), 3
NORTH_EAST (lackey.Region attribute), 3
NORTH_MID (lackey.Region attribute), 3
NORTH_WEST (lackey.Region attribute), 3
NUM0 (lackey.Key attribute), 19
NUM1 (lackey.Key attribute), 19
NUM2 (lackey.Key attribute), 19
NUM3 (lackey.Key attribute), 19
NUM4 (lackey.Key attribute), 19
NUM5 (lackey.Key attribute), 19
NUM6 (lackey.Key attribute), 19
NUM7 (lackey.Key attribute), 19
NUM8 (lackey.Key attribute), 19
NUM9 (lackey.Key attribute), 19
NUM_LOCK (lackey.Key attribute), 19

O
ObserveMinChangedPixels (lackey.SettingsMaster attribute), 22
observe() (lackey.Region method), 9
observeInBackground() (lackey.Region method), 9
ObserveScanRate (lackey.SettingsMaster attribute), 22
OcrDataPath (lackey.SettingsMaster attribute), 22

L
LEFT (lackey.Button attribute), 18
LEFT (lackey.Key attribute), 19
LEFT (lackey.Mouse attribute), 15
left() (lackey.Location method), 15
left() (lackey.Region method), 8
leftAt() (lackey.Region method), 8
LL (lackey.Region attribute), 3
Location (class in lackey), 14
log() (lackey.DebugMaster method), 21
LogTime (lackey.SettingsMaster attribute), 22

H
hasEvents() (lackey.Region method), 7
hasMoved() (lackey.Mouse method), 16
hasObserver() (lackey.Region method), 7
hasPrompt() (lackey.Screen method), 14
hasWindow() (lackey.App method), 17
highlight() (lackey.Region method), 8
history() (lackey.DebugMaster method), 20
HOME (lackey.Key attribute), 19
hover() (lackey.Location method), 15
hover() (lackey.Region method), 8

I
ImagePath (lackey.SettingsMaster attribute), 22
info() (lackey.DebugMaster method), 21
InfoLogs (lackey.SettingsMaster attribute), 22
input() (in module lackey), 23
inputText() (in module lackey), 24
INSERT (lackey.Key attribute), 19
inside() (lackey.Key attribute), 19
intersection() (lackey.Region method), 8
isChanged() (lackey.Region method), 8
isImagePattern() (lackey.Pattern method), 12
isObserving() (lackey.Region method), 8
isRasterValid() (lackey.Region method), 8
isRegionValid() (lackey.Region method), 8
isRunning() (lackey.App method), 18
isValid() (lackey.App method), 18
isValid() (lackey.Pattern method), 12

K
Key (class in lackey), 18
Keyboard (class in lackey), 17
keyDown() (lackey.Keyboard method), 17
keyDown() (lackey.Region method), 8
KeyModifier (class in lackey), 20
keyUp() (lackey.Keyboard method), 17
keyUp() (lackey.Region method), 8

Index 31
off() (lackey.DebugMaster method), 21
offset() (lackey.Location method), 15
offset() (lackey.Region method), 9
on() (lackey.DebugMaster method), 21
onAppear() (lackey.Region method), 9
onChange() (lackey.Region method), 9
onVanish() (lackey.Region method), 9
open() (lackey.App class method), 18

P
PAGE_DOWN (lackey.Key attribute), 20
PAGE_UP (lackey.Key attribute), 20
paste() (lackey.Region method), 9
Pattern (class in lackey), 12
PAUSE (lackey.Key attribute), 20
pause() (lackey.App class method), 18
PlatformManagerWindows (in module lackey), 20
popAsk() (in module lackey), 23
popat() (in module lackey), 23
popError() (in module lackey), 23
popFile() (in module lackey), 24
popup() (in module lackey), 23
PopupLocation (lackey.SettingsMaster attribute), 22
primaryScreen (lackey.Screen attribute), 14
PRINTSCREEN (lackey.Key attribute), 20
PROMPT (lackey.Region attribute), 3

R
Region (class in lackey), 3
resetMonitors() (lackey.Screen method), 14
resetPrompt() (lackey.Region method), 9
RETRY (lackey.Region attribute), 3
RIGHT (lackey.Button attribute), 18
RIGHT (lackey.Key attribute), 20
RIGHT (lackey.Mouse attribute), 16
right() (lackey.Location method), 15
right() (lackey.Region method), 10
rightAt() (lackey.Region method), 10
rightClick() (lackey.Region method), 10
RR (lackey.Region attribute), 4

S
saveCapture() (lackey.Screen method), 14
saveLastScreenImage() (lackey.Region method), 10
saveScreenCapture() (lackey.Region method), 10
Screen (class in lackey), 13
SCROLL_LOCK (lackey.Key attribute), 20
select() (in module lackey), 24
selectRegion() (lackey.Screen method), 14
SEPARATOR (lackey.Key attribute), 20
setActive() (lackey.Region method), 10
setAutoWaitTimeout() (lackey.Region method), 10
setBottomLeft() (lackey.Region method), 10
setBottomRight() (lackey.Region method), 10
setBundlePath() (in module lackey), 23
setCenter() (lackey.Region method), 10
setClipboard() (lackey.App class method), 18
setCols() (lackey.Region method), 10
setFilename() (lackey.App class method), 18
setFindFailedHandler() (lackey.Region method), 10
setFindFailedResponse() (lackey.Region method), 10
setH() (lackey.Region method), 10
setImage() (lackey.Pattern method), 12
setImageMissingHandler() (lackey.Region method), 10
setInactive() (lackey.Region method), 10
setLocation() (lackey.Location method), 15
setLocation() (lackey.Region method), 10
setLogFile() (lackey.DebugMaster method), 21
setLogger() (lackey.DebugMaster method), 21
setLoggerAction() (lackey.DebugMaster method), 21
setLoggerAll() (lackey.DebugMaster method), 21
setLoggerDebug() (lackey.DebugMaster method), 21
setLoggerError() (lackey.DebugMaster method), 21
setLoggerInfo() (lackey.DebugMaster method), 21
setLoggerNoPrefix() (lackey.DebugMaster method), 21
setLoggerUser() (lackey.DebugMaster method), 21
setObserveScanRate() (lackey.Region method), 11
setRaster() (lackey.Region method), 11
setRect() (lackey.Region method), 11
setRepeatWaitTime() (lackey.Region method), 11
setROI() (lackey.Region method), 10
setRows() (lackey.Region method), 11
setShowActions() (in module lackey), 23
setSize() (lackey.Region method), 11
setThrowException() (lackey.Region method), 11
SettingsMaster (class in lackey), 22
setTopLeft() (lackey.Region method), 11
setTopRight() (lackey.Region method), 11
setUsing() (lackey.App method), 18
setW() (lackey.Region method), 11
setWaitScanRate() (lackey.Region method), 11
setX() (lackey.Region method), 11
setY() (lackey.Region method), 11
SHIFT (lackey.Key attribute), 20
SHIFT (lackey.KeyModifier attribute), 20
ShowActions (lackey.SettingsMaster attribute), 22
showMonitors() (lackey.Screen class method), 14
showScreens() (lackey.Region method), 11
showTarget() (lackey.Screen method), 14
similar() (lackey.Pattern method), 12
SKIP (lackey.Region attribute), 4
sleep() (in module lackey), 23
SlowMotionDelay (lackey.SettingsMaster attribute), 22
SOUTH (lackey.Region attribute), 4
SOUTH_EAST (lackey.Region attribute), 4
SOUTH_MID (lackey.Region attribute), 4
SOUTH_WEST (lackey.Region attribute), 4
SPACE (lackey.Key attribute), 20
stopObserver() (lackey.Region method), 11

T
TAB (lackey.Key attribute), 20
targetOffset() (lackey.Pattern method), 12
text() (lackey.Region method), 11
translate() (lackey.Location method), 15
TT (lackey.Region attribute), 4
type() (lackey.Keyboard method), 17
type() (lackey.Region method), 11
TypeDelay (lackey.SettingsMaster attribute), 22

U
union() (lackey.Region method), 11
unzip() (in module lackey), 23
UP (lackey.Key attribute), 20
up() (lackey.Mouse method), 16
user() (lackey.DebugMaster method), 21
userCapture() (lackey.Screen method), 14
UserLogPrefix (lackey.SettingsMaster attribute), 22
UserLogs (lackey.SettingsMaster attribute), 22
UserLogTime (lackey.SettingsMaster attribute), 22

W
wait() (lackey.Region method), 12
waitForWindow() (lackey.App method), 18
WaitScanRate (lackey.SettingsMaster attribute), 22
waitVanish() (lackey.Region method), 12
WEST (lackey.Region attribute), 4
WEST_MID (lackey.Region attribute), 4
wheel() (lackey.Mouse method), 16
wheel() (lackey.Region method), 12
WHEEL_DOWN (lackey.Mouse attribute), 16
WHEEL_UP (lackey.Mouse attribute), 16
WIN (lackey.Key attribute), 20
WIN (lackey.KeyModifier attribute), 20
window() (lackey.App method), 18
write() (lackey.Region method), 12