

---

# DDRescue-GUI Developer Documentation

*Release 2.2.0*

Hamish McIntyre-Bhatty

Jul 31, 2024



## CONTENTS

1	Documentation for the main executable file (DDRescue_GUI.py)	3
2	Documentation for the unit tests runner file (tests.py)	11
3	Documentation for the unit tests package (Tests)	13
4	Documentation for the tools package (Tools)	15
5	Documentation for the core tools module in the tools package (Tools/core.py)	17
6	Documentation for the mounting tools module in the tools package (Tools/mount_tools.py)	21
7	Documentation for the ddrescue tools package in the tools package (Tools/DDRescueTools)	25
8	Documentation for the setup module in the ddrescue tools package in the tools package (Tools/DDRescueTools/setup.py)	27
9	Documentation for the decorators module in the ddrescue tools package in the tools package (Tools/DDRescueTools/decorators.py)	29
10	Documentation for the allversions module in the ddrescue tools package in the tools package (Tools/DDRescueTools/allversions.py)	31
11	Documentation for the ddrescue tools for ddrescue version 1.14 in the ddrescue tools package in the tools package (Tools/DDRescueTools/one_point_fourteen.py)	33
12	Documentation for the ddrescue tools for ddrescue version 1.18 in the ddrescue tools package in the tools package (Tools/DDRescueTools/one_point_eighteen.py)	35
13	Documentation for the ddrescue tools for ddrescue version 1.20 in the ddrescue tools package in the tools package (Tools/DDRescueTools/one_point_twenty.py)	37
14	Documentation for the ddrescue tools for ddrescue version 1.21 in the ddrescue tools package in the tools package (Tools/DDRescueTools/one_point_twenty_one.py)	39
15	Documentation for the ddrescue tools for ddrescue version 1.22 in the ddrescue tools package in the tools package (Tools/DDRescueTools/one_point_twenty_two.py)	41
16	Indices and tables	43

Python Module Index	45
Index	47

Contents:



## DOCUMENTATION FOR THE MAIN EXECUTABLE FILE (DDRESCUE\_GUI.PY)

This is the main script that you use to start DDRescue-GUI.

**class** ddrescue\_gui.DDRescue\_GUI.**BackendThread**(*parent*)

Handles getting input from ddrescue during a recovery, and forwards it back to the GUI thread as required.

**process\_line**(*line*)

Process a given line to get ddrescue's current status and recovery information and send it to the GUI Thread

**run**()

Main body of the thread, started with self.start().

**class** ddrescue\_gui.DDRescue\_GUI.**CustomTextCtrl**(*parent, wx\_id, value, style*)

A custom wx.TextCtrl that provides features that are broken on Linux and Cygwin.

**Features:** A version of XYToPosition() that fixes a bug on Linux and Cygwin. carriage\_return(): Handles carriage returns correctly. up\_one\_line(): Moves insertion point up one line.

**XYToPosition**(*column, row*)

A custom version of wx.TextCtrl.XYToPosition() that fixes a bug on Linux and Cygwin.

**Args:**

**column (int).** The column we want to get the integer position for.

**row (int).** The row we want to get the integer position for.

**Returns:** int. The position.

---

**Note:** This is required on Linux/Cygwin because the built-in one has a quirk: when you're at the end of the text, it always returns -1.

---

**add\_line**(*data, crs, uols, char\_number*)

Adds a new line to the custom output box. Also handles calling carriage\_return() and up\_one\_line() when required. Receives the data chunks and other information from the update method.

**Args:**

**data (string).** The chunk of text to add to the output box.

**crs (list).** A list of character numbers where the character is a carriage return.

**uols (list).** As above, for up-one-line sequences.

**char \_number (int).** The character number we are at in the line (the character after the last character in our chunk of text).

**carriage\_return()**

Handles carriage returns in output. This is done by going back to the last newline in the box - any new text will now overwrite what is there.

**up\_one\_line()**

Handles (control sequence to go up one line) in the output. This is done by moving the insertion point so we are up one line, but in the same column (if possible).

**update(*line*)**

Append the given line to the contents of the output box. Counts carriage returns and up-one-lines so that an auxiliary method (`add_line`) can handle them.

**Args:** *line* (string). The line to add.

**class ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow(*parent*)**

DDRescue-GUI's disk information window.

**bind\_events()**

Bind all events for DiskInfoWindow

**create\_widgets()**

Create all widgets for DiskInfoWindow

**get\_diskinfo(*event=None*)**

Call the thread to get Disk info, disable the refresh button, and start the throbber

**on\_exit(*event=None*)**

Exit DiskInfoWindow

**on\_size(*event=None*)**

Auto resize the `list_ctrl` columns

**receive\_diskinfo(*info*)**

Get Disk data, call `self.update_list_ctrl()`, and then call `MainWindow().update_file_choices()` to refresh the file choices with the new info.

**Args:** *info* (dict). The new disk information.

**setup\_sizers()**

Set up the sizers for DiskInfoWindow

**update\_list\_ctrl(*event=None*)**

Update the list control

**class ddrescue\_gui.DDRescue\_GUI.ElapsedTimeThread(*parent*)**

Keeps track of elapsed time during a recovery. A separate thread is used for this because `wx.Timer` wasn't working on macOS, and the `BackendThread` blocks if `ddrescue` pauses.

**run()**

Main body of the thread, started with `self.start()`.

**class ddrescue\_gui.DDRescue\_GUI.FinishedWindow(*parent, disk\_capacity, recovered\_data*)**

This is displayed after a recovery is finished/aborted. Used to provide the user w/ options to restart the GUI, mount the output file, or close the GUI.

**bind\_events()**

Bind all events for FinishedWindow.

**create\_buttons()**

Create all buttons for FinishedWindow.



**create\_text()**

Create all text for FinishedWindow.

**on\_browse(event=None)**

Open the file viewer and browse the mounted volume.

**on\_exit(event=None)**

Close FinishedWindow and trigger closure of MainWindow.

**on\_mount(event=None)**

Triggered when mount button is pressed, used to initiate mounting the output file/device.

**restart(event=None)**

Close FinishedWindow and call MainWindow().restart() to re-display and reset MainWindow.

**setup\_sizers()**

Set up all sizers for FinishedWindow.

**class ddrescue\_gui.DDRescue\_GUI.GetDiskInformation(parent)**

Used to get disk information without blocking the GUI thread. Calls parent.receive\_diskinfo when info has ben retrieved.

**get\_info()**

Get disk information as a privileged user.

**Returns:**

**dict.** If successful: The disk information. If unsuccessful: An empty dictionary.

**run()**

Use GetDevInfo module to get disk information.

**class ddrescue\_gui.DDRescue\_GUI.MainWindow**

DDRescue-GUI's main window.

**bind\_events()**

Bind all events for MainWindow

**check\_for\_updates(event=None, starting\_up=False)**

Check for updates using the plist-formatted update file on my website. If on startup, only display info to the user if there was an update. Otherwise (aka requested by user), always display the information.

**Args:**

**starting\_up[=True] (boolean).** If the GUI is starting up, specify True, otherwise leave unspecified.

**create\_buttons()**

Create all buttons for MainWindow

**create\_choice\_boxes()**

Create all choiceboxes for MainWindow

**create\_menus()**

Create the menus

**create\_other\_widgets()**

Create all other widgets for MainWindow

**create\_text()**

Create all text for MainWindow

### **define\_vars()**

Defines some variables used elsewhere in this class/instance.

### **file\_choice\_handler(*\_type*, *user\_selection*, *default\_dir*, *wildcard*, *style*)**

Handle file dialogs for `set_input_file`, `set_output_file`, and `set_map_file`.

**Args:**

**`_type` (string).** The type of file we're handling. "Input", "Output", or "Map".

**`user_selection` (string):** The option the user selected in the choice box.

**`default_dir` (string):** The default directory any file dialogs are to use.

**`wildcard` (string):** The wildcard that any file dialogs are to use.

**`style` (int):** The style that any file dialogs are to use.

### **focus\_on\_control\_button(*event=None*)**

Focus on the control button instead of the TextCtrl, and reset the insertion point back after 30 milliseconds, preventing the user from changing the insertion point and messing the formatting up.

### **get\_confirm\_text()**

Generate recovery confirmation text to make doubly sure the right devices are selected. This can be logged and put in a message dialog.

**Returns:** String. The confirmation text.

### **get\_diskinfo(*event=None*)**

Call the thread to get Disk info, disable the update button, and start the throbber

### **handle\_no\_mapfile(*key*, *choice\_box*)**

Handles when the user selects not to have a mapfile.

**Args:** `key` (string): The unique key used to identify the output file. `choice_box` (wx.Choice): The output choice box.

### **handle\_outputfile\_special\_cases(*key*, *choice\_box*)**

Handles special cases for the output choice box.

**Args:** `key` (string): The unique key used to identify the output file. `choice_box` (wx.Choice): The output choice box.

### **handle\_user\_file\_selection(*\_type*, *key*, *user\_selection*, *paths*, *choice\_box*)**

Handles user file selection for the main settings choiceboxes.

**Args:** `_type` (string): The type of file we're setting (Input, Output, or Map). `key` (string): The unique key used to identify the output file. `user_selection` (string): The user's selected path/file. `paths` (dict): The custom paths defined for this type of file. `choice_box` (wx.Choice): The output choice box.

### **make\_status\_bar()**

Create and set up a statusbar

### **on\_abort()**

Abort the recovery.

### **on\_about(*event=None*)**

Show the about box.

### **on\_control\_button(*event=None*)**

Handle events from the control button, as its purpose changes during and after recovery. Call `self.on_abort()` when clicked during a recovery. Call `self.on_start()` otherwise.

**on\_detailed\_info**(*event=None*)

Show/Hide the detailed info, and rotate the arrow next to the text label.

**on\_exit**(*event=None, just\_finished\_recovery=False*)

Exit DDRescue-GUI, if certain conditions are met (for example we aren't in the middle of a recovery). Also offer to save the log file for debugging / error-reporting purposes.

**Args:**

**just\_finished\_recovery (bool).**

**True - Display FinishedWindow if user cancels** the exit attempt.

**False - The default, do nothing if user cancels** the exit attempt.

**on\_mount**(*event=None*)

When the user asks to mount a file, handle this and show FinishedWindow in order to carry out the request.

**on\_recovery\_ended**(*result, disk\_capacity, recovered\_data, return\_code=None*)

Called by the backend thread to show FinishedWindow and update the main window when a recovery is completed or aborted by the user, or when a recovery errors out for some reason.

**Args:**

**result (string). The reason why the recovery ended. Used to let the user know** what is happening. Values are "NoInitialStatus", "BadReturnCode", and "Success".

**disk\_capacity (string).** The capacity of the input file or disk. **recovered\_data (string).** The amount of data we recovered.

**return\_code[=None] (int).** GNU ddrescue's return code. Useful if the recovery failed for some reason.

**on\_session\_end**(*event*)

Attempt to veto e.g. a shutdown/logout event if recovering data.

**on\_size**(*event=None*)

Auto resize the list\_ctrl columns when the window is resized.

**on\_start**()

Check the settings, prepare to start ddrescue, unmount the input file if needed, and start the backend thread.

**on\_terminal\_output**(*event=None*)

Show/Hide the terminal output, and rotate the arrow next to the text label.

**prompt\_to\_kill\_ddrescue**()

Prompts the user to try killing ddrescue again if it's not exiting. This sometimes happens if the system is overloaded, or if a disk is taking a very long time to timeout/fail a read operation.

**receive\_diskinfo**(*info*)

Get new Disk info, stop the throbber and call the function that updates the choiceboxes for input and output file selection.

**restart**()

Restart and reset MainWindow, so MainWindow is as it was when DDRescue-GUI was started.

**save\_debug\_log**(*event=None*)

Save DDRescue-GUI's debug log.

**set\_input\_file**(*event=None*)

Get the input file/Disk by calling self.file\_choice\_handler.

**set\_map\_file**(*event=None*)  
 Get the map file position/name by calling self.file\_choice\_handler.

**set\_output\_file**(*event=None*)  
 Get the output file/Disk by calling self.file\_choice\_handler.

**set\_progress\_bar\_range**(*\_range*)  
 Set the progress bar's range.  
 Args: *\_range* (int). The range to set the progress bar to use.

**set\_vars**()  
 Set some essential variables

**setup\_sizers**()  
 Setup sizers for MainWindow

**show\_dev\_info**(*event=None*)  
 Show the Disk Information Window.

**show\_inspector**(*event*)  
 Shows the wxPython inspection tool.

**show\_privacy\_policy**(*event=None*)  
 Show the Privacy Policy Window

**show\_settings**(*event=None*)  
 Show the settings Window, but only if input and output files have already been selected.

**show\_userguide**(*event=None*)  
 Open a web browser and show the user guide.

**update\_average\_read\_rate**(*average\_read\_rate*)  
 Update the average read rate info.  
 Args: *average\_read\_rate* (string). The average read rate.

**update\_current\_read\_rate**(*current\_read\_rate*)  
 Update the current read rate info.  
 Args: *current\_read\_rate* (string). The current read rate.

**update\_error\_size**(*error\_size*)  
 Update the error size info.  
 Args: *error\_size* (string). The amount of unreadable data so far.

**update\_file\_choices**()  
 Update the disk entries in the choiceboxes

**update\_input\_pos**(*input\_pos*)  
 Update the input position info.  
 Args:  
     *input\_pos* (string). The current position in the input file or device.

**update\_num\_errors**(*num\_errors*)  
 Update the num errors info.  
 Args: *num\_errors* (string). The number of read errors so far.

**update\_output\_pos**(*output\_pos*)  
 Update the output position info.  
 Args:

**output\_pos (string).** The current position in the output file or device.

**update\_progress(*recovered\_data*, *disk\_capacity*)**

Update the progress bar and the title. Do nothing if disk capacity is unknown.

**Args:**

**recovered\_data (int).** The amount of data currently recovered (units vary based on disk size).

**disk\_capacity (int).** The capacity (or size) of the input file or disk.

**update\_recovered\_data(*recovered\_data*)**

Update the recovered data info.

**Args:** *recovered\_data* (string). The amount of data recovered so far.

**update\_status\_bar(*message*)**

Update the status bar with a new message.

**Args:** *message* (string). The message to set the status bar to.

**update\_time\_elapsed(*time\_elapsed*)**

Update the time elapsed text.

**Args:**

**time\_elapsed (string).** The label to use for the time elapsed text.

**update\_time\_remaining(*time\_left*)**

Update the time remaining text.

**Args:**

**time\_remaining (string).** The label to use for the time remaining text.

**update\_time\_since\_last\_read(*last\_read*)**

Update the time since last successful read info.

**Args:**

**last\_read (string).** The amount of time that has passed since ddrescue successfully read any data from the input file.

**class ddrescue\_gui.DDRescue\_GUI.MyApp(*redirect=False*, *filename=None*, *useBestVisual=False*, *clearSigInt=True*)**

The wxPython app. Must be declared for application to work. This is how the application is started.

**MacReopenApp()**

Called when the doc icon is clicked, shows the top-level window again even if it's minimised. Makes the GUI work in a more intuitive way on macOS.

**OnInit()**

Used to show the splash screen, which then starts the rest of the application.

**class ddrescue\_gui.DDRescue\_GUI.PrivPolWindow(*parent*)**

DDRescue-GUI's privacy policy window.

**bind\_events()**

Bind events so we can close this window.

**create\_widgets()**

Create all widgets for PrivPolWindow

**on\_close(*event=None*)**

Close PrivPolWindow.

```
setup_sizers()
    Set up sizers for PrivPolWindow

class ddrescue_gui.DDRescue_GUI.SettingsWindow(parent)
    DDRescue-GUI's settings window

bind_events()
    Bind all events for SettingsWindow.

create_buttons()
    Create all buttons for SettingsWindow

create_check_boxes()
    Create all CheckBoxes for SettingsWindow, and set their default states (all unchecked)

create_choice_boxes()
    Create all ChoiceBoxes for SettingsWindow, and call self.set_default_recovery_settings()

create_text()
    Create all text for SettingsWindow

save_options(event=None)
    Save all options, and exit SettingsWindow.

set_best_recovery_settings(event=None)
    Set selections for the Choiceboxes to best recovery settings.

set_default_recovery_settings(event=None)
    Set selections for the Choiceboxes to default settings.

set_fast_recovery_settings(event=None)
    Set selections for the Choiceboxes to fast recovery settings.

set_soft_run(event=None)
    Set up SettingsWindow based on the value of self.no_split_check_box (the “do soft run” Check-
    Box).

setup_options()
    Set all options in the window so we remember them if the user checks back

setup_sizers()
    Set up all sizers for SettingsWindow.

class ddrescue_gui.DDRescue_GUI.ShowSplash(parent=None)
    A simple class used to display the splash screen on startup. After that, it starts the rest of the
    application.

on_exit(event=None)
    Close the splash screen and start MainWindow.

Args:
    event[=None] (object). The event object passed by wxPython when the splash times
    out.

ddrescue_gui.DDRescue_GUI.usage()
    Outputs information on cmdline options for the user.
```

## DOCUMENTATION FOR THE UNIT TESTS RUNNER FILE (TESTS.PY)

This file is used to start the test suites for DdRescue-GUI.

```
ddrescue_gui.tests.usage()  
    Outputs usage information
```





## DOCUMENTATION FOR THE UNIT TESTS PACKAGE (TESTS)

This is the Tests package.



## DOCUMENTATION FOR THE TOOLS PACKAGE (TOOLS)

This is the Tools package.



## DOCUMENTATION FOR THE CORE TOOLS MODULE IN THE TOOLS PACKAGE (TOOLS/CORE.PY)

This is the tools package for DDRescue-GUI.

**class ddrescue\_gui.Tools.core.AuthWindow**

A simple authentication dialog that is used when elevated privileges are required. Until version 2.0.0, this was used to start the GUI, but since that release, privileges are only escalated when required to improve security.

This is used to pre-authenticate on macOS if needed, before running a privileged task with sudo.

**bind\_events()**

Bind all events for AuthenticationWindow

**create\_buttons()**

Create all buttons for AuthenticationWindow

**create\_other\_widgets()**

Create all other widgets for AuthenticationWindow

**create\_text()**

Create all text items for AuthenticationWindow.

**on\_auth\_attempt(event=None)**

Check the password is correct. If not, then either warn the user to try again. If so, exit as all we need to do is pre-authenticate on macOS.

**Args:** event. The event object passed by wxpython (optional).

**on\_exit(event=None)**

Close AuthWindow() and exit

**run()**

Preauthenticates macOS users with the auth dialog. If we are already pre-authenticated, just return immediately.

**setup\_sizers()**

Setup sizers for AuthWindow

**test\_auth()**

Check if we have cached authentication.

**Returns:**

**bool.** True = We have cached authentication. False = We don't.

**ddrescue\_gui.Tools.core.change\_units(number\_to\_change, current\_unit, required\_unit)**

Convert data so it uses the correct unit of measurement.

**Args:**

**number\_to\_change (int).** The number we wish to change the units for.

**current\_unit (string).** The current unit of this number. **required\_unit (string).** The required unit for this number.

**Returns:** tuple(int, string).

1st element: The number's value in its new unit. 2nd element: The new unit.

**ddrescue\_gui.Tools.core.create\_unique\_key(dictionary, data, length)**

Create a unique dictionary key.

The unique key is created by adding a number on the the end of the given data, while keeping it at the correct length. The key will also start with '...' if the data was longer than the specified length.

**Args:**

**dictionary (dict).** The dictionary that the key will be stored in. This is needed to check the uniqueness of the keys - we will keep generating new ones until we arrive at a unique one.

**data (string).** The data that we need to create a key for.

**length (int).** The maximum length of the key.

**Returns:** string. The unique key.

**ddrescue\_gui.Tools.core.determine\_ddrescue\_version()**

Used to determine the version of ddrescue installed on the system, or (for macOS and Windows) bundled with the GUI.

Handles -pre and -rc versions too, by stripping that information from the version string and warning the user (not doing so would cause errors in other parts of DDRescue-GUI).

**Returns:** string. The ddrescue version present on the system.

**ddrescue\_gui.Tools.core.determine\_getdevinfo\_version()**

Used to determine the version of getdevinfo installed on the system, or (for macOS and Windows) bundled with the GUI.

**Returns:** string. The getdevinfo version present on the system.

**ddrescue\_gui.Tools.core.emergency\_exit(msg)**

Handle emergency exits. Warn the user, log the error, save the log file, and exit to terminal with the given message.

**Args:**

**msg (string).** A description of the unrecoverable error that was encountered.

**Warning:** Calling this function will exit DDRescue-GUI immediately after warning the user to save a log file.

**ddrescue\_gui.Tools.core.find\_ddrescue()**

Attempts to find GNU ddrescue, and ends the program if it couldn't be found.

**ddrescue\_gui.Tools.core.get\_helper(cmd)**

Figure out which helper script to use for this command.

**Args:** cmd (string). The command(s) about to be run.

**Returns:**

string. “pkexec” + <the helper script needed>

- the command(s) to run.

`ddrescue_gui.Tools.core.get_mount_point(partition)`

Returns the mountpoint of the given partition, if any.

**Args:** partition (string). The partition to find the mount point of.

**Returns:** Multiple types.

**String:** The mount point of the partition, if it was mounted.

**None:** Returned when mount point was not found.

`ddrescue_gui.Tools.core.is_mounted(partition, mount_point=None)`

Checks if the given partition is mounted.

**Args:** partition (string). The partition to check.

**mount\_point[=None] (string).** If specified, check that partition is mounted at this mount point. Otherwise, just check that it is mounted somewhere.

**Returns:** bool.

True = The partition is mounted. False = The partition is not mounted.

`ddrescue_gui.Tools.core.is_partition(disk, disk_info)`

Check if the given disk is a partition.

**Args:** disk (string). The disk to check. disk\_info (dict). The disk info dictionary containing information gathered with GetDevInfo.

**Returns:**

**boolean.** True - The disk is a partition. False - The disk is not a partition.

`ddrescue_gui.Tools.core.mount_disk(partition, mount_point, options='')`

Mounts the given partition at the given mount point.

**Args:** partition (string). The partition to mount. mount\_point (string). The path where the partition is to be mounted.

**options[=""] (string).** Any options to pass to the mount command. If not specified, no options are passed.

**Returns:** Multiple types.

**boolean False:** If another filesystem was in the way at the specified mount point and it could not be unmounted.

**int.**

**0 - Success, or partition already mounted at** that mount point.

Anything else - Error, return value from mount command.

`ddrescue_gui.Tools.core.read(cmd, testing=False)`

Read the cmd’s output character by character. Also make sure everything is converted to unicode. Break lines by the newline and (carriage return) characters. Also handle null characters by removing them from the output.

**Args:** cmd. The subprocess object that represents the command.

**testing[=False].** Used during unit tests, disables some of the cleanup done to the output. **Do not use in production.**

**Returns:**

**list.** A list where each line in the (cleaned up) output is a new item in the list.

`ddrescue_gui.Tools.core.send_notification(msg)`

Send a notification, with the given message.

**Args:** msg (string). The message to display in the notification.

`ddrescue_gui.Tools.core.start_process(cmd, return_output=False, privileged=False)`

Start a given process, and return the output and return value if needed.

**Args:** cmd (string). The command(s) to run.

**return\_output[=False]** Whether to return the output or not. If not specified, the default is False.

**privileged[=False]** Whether to execute the command(s) with elevated privileges or not. If not specified the default is False.

**Returns:** May return multiple types:

**int.** If `return_output` is not specified or set to False, return the return value of the command(s).

**tuple(int, string).** Otherwise, return a tuple with the return value, and then a string with new lines delimited by newline characters.

`ddrescue_gui.Tools.core.unmount_disk(disk)`

Unmount the given disk.

**Args:** disk (string). The disk to unmount.

**Returns:**

**int.** 0 - Success, or disk not mounted. Anything else - Error, return value from unmount command.



## DOCUMENTATION FOR THE MOUNTING TOOLS MODULE IN THE TOOLS PACKAGE (TOOLS/MOUNT\_TOOLS.PY)

This is the destination file mount tools module in the tools package for DDRescue-GUI.

**class ddrescue\_gui.Tools.mount\_tools.Core**

This class contains core methods used on both Linux and macOS

**classmethod mount\_output\_file()**

Mount the output file in SETTINGS["OutputFile"].

**Returns:**

**boolean.** True - Success False - Failed

**classmethod reset()**

Resets the state of this class, and triggers reset of the Linux and Mac classes.

**classmethod unmount\_output\_file()**

Unmount the output file.

**Returns:**

**boolean.** True - Success False - Failed

**class ddrescue\_gui.Tools.mount\_tools.Linux**

Linux-specific stuff for mounting the output file.

**classmethod determine\_output\_file\_type(output\_file)**

Determines output File Type (partition or device).

**Args:** output\_file (str): The output file or device to determine the type of.

**Returns:** tuple(string, bool).

**1st element: The type of the output file. "Partition", "Device", or "LVM".**

**2nd element: True - success, False - failed.**

**classmethod get\_volumes\_lvm(output\_file)**

Gets a list of volumes on the given output file or device name. This method expects the given file or device to be an LVM Physical Volume.

**Args:** output\_file (str): The output file or device to get volumes for.

**Returns.** list. The volumes that were found in human-readable form.

**classmethod get\_volumes\_std\_device(output\_file)**

Gets a list of volumes on the given output file or device name. This method expects the given file or device to be a standard device.

**Args:** output\_file (str): The output file or device to get volumes for.

**Returns.** list. The volumes that were found in human-readable form.

**classmethod mount\_device(*output\_file*)**

Mounts the given output file or device, expecting it to be a standard device or another kind of container for volumes - LVM.

**Args:** *output\_file* (str). The device or file to mount.

**Returns:**

**Boolean.** True - Success False - Failure

**classmethod mount\_partition(*partition*)**

Mounts the given file or device name as a single volume or partition.

**Args:** *partition* (str): The file or device to mount.

**Returns:**

**Boolean.** True - Success False - Failed

**classmethod reset()**

Resets the state of this class to defaults.

**classmethod unmount\_output\_file(*output\_file*)**

Unmounts the output file or device. Handles partitions, devices, and LVM disks.

**Args:** *output\_file* (str). The device or file to unmount.

**Returns:**

**Boolean.** True - Success False - Failed

**class ddrescue\_gui.Tools.mount\_tools.Mac**

Macos-specific stuff for mounting the output file.

**classmethod attach\_file(*output\_file*)**

Attaches the given output file to the system as a read-only device.

**Args:** *output\_file* (str). The output file to attach.

**Returns:**

**tuple. Elements:** 1 - int. The return value from hdiutil attach. 2 - str. The device name of the file, or None if attaching failed.

**classmethod determine\_output\_file\_type(*output\_file*)**

Determines output File Type (partition or device)..

**Returns:** tuple(string, bool).

**1st element: The type of the output file. “Partition”, “Device”, “CD”, “APFSStore”, “APFSContainer” or “APFSVolume”.**

**2nd element:** True - success, False - failed.

**classmethod get\_device\_name(*output*)**

Get the device name of an output file, given output from hdiutil attach -plist.

**Args:**

**output (string).** Output from “hdiutil attach -plist”, the command used to mount the output file.

**Returns:** tuple(<inconsistent types>).

**1st element:** The device name of the output file eg “/dev/disk5”, or None if unable to determine it.

**2nd element:** True (boolean) if successful in determining device name and mount point. Otherwise, a string describing the error eg “UnicodeError”.

**classmethod `get_volumes_apfs(output_file)`**

Finds volumes contained by APFS containers.

**Args:** `output_file` (str). The output file or device to investigate.

**Returns.** list. The volumes that were found in human-readable form.

**classmethod `get_volumes_std_device(output_file, cdimage=False)`**

Finds volumes contained by standard devices.

**Args:** `output_file` (str). The output file or device to investigate.

`cdimage[=False]` (bool). Whether or not we are finding volumes on a CD device/image.

**Returns.** list. The volumes that were found in human-readable form.

**classmethod `mount_device(output_file)`**

Mount the given device or file. This is expected to be a standard device or other container of volumes (eg an APFS container).

**Args:** `output_file` (str). The device or file to mount.

**Returns:**

**Boolean.** True - Success False - Failure

**classmethod `mount_partition(partition, attach=False)`**

Mounts the given partition, also attaching the file if needed.

**Args:** `partition` (str). The partition or file to mount.

`attach[=False]` (bool). Whether to attach the file first.

**Returns:**

**boolean.** True - Success False - Failed

**classmethod `reset()`**

Resets the state of this class to defaults. No action needed, just here to retain compatibility.

**classmethod `run_hdiutil(options)`**

Runs hdiutil on behalf of the rest of the program when called. Tries to handle and fix hdiutil errors (e.g. ‘Resource Temporarily Unavailable’) if they occur.

**Args:** `options` (string). All of the options to pass to hdiutil.

**Returns:** tuple(int, string).

1st element: The return value from hdiutil.

2nd element: The output from hdiutil.

**classmethod `unmount_output_file(devicename)`**

Unmounts the given device. Can be used for output files as well, but needs to be given the device associated with them.

**Args:** `devicename` (str). The device to unmount.

**Returns:**

**Boolean.** True - Success False - Failed



## DOCUMENTATION FOR THE DDRESCUE TOOLS PACKAGE IN THE TOOLS PACKAGE (TOOLS/DDRESCUETOOLS)

These are the ddrescue tools.



## DOCUMENTATION FOR THE SETUP MODULE IN THE DDRESCUE TOOLS PACKAGE IN THE TOOLS PACKAGE (TOOLS/DDRESCUETOOLS/SETUP.PY)

Used to set up the GUI to use the correct version of tools for the user's version of ddrescue.

`ddrescue_gui.Tools.DDRescueTools.setup.setup_for_ddrescue_version(ddrescue_version)`

Selects and returns a list of the correct functions for our version of ddrescue.

**Args:**

**ddrescue\_version (str):** The version of ddrescue installed on the system. eg "1.26".

**Returns:**

**list.** A list of all the functions that are designed to work with this ddrescue version.





## DOCUMENTATION FOR THE DECORATORS MODULE IN THE DDRESCUE TOOLS PACKAGE IN THE TOOLS PACKAGE (TOOLS/DDRESCUETOOLS/DECORATORS.PY)

Decorators for DDRescue tools

`ddrescue_gui.Tools.DDRescueTools.decorators.define_versions(function)`

Reads the function docstring to find the ddrescue versions the function supports. This is used on all of the tools in the modules in the DDRescueTools package.

This information is saved in the function's SUPPORTEDVERSIONS attribute.

**Args:**

**function.** The function object that we are creating the attribute for.



## DOCUMENTATION FOR THE ALLVERSIONS MODULE IN THE DDRESCUE TOOLS PACKAGE IN THE TOOLS PACKAGE (TOOLS/DDRESCUETOOLS/ALLVERSIONS.PY)

Tools for all versions of ddrescue.

`ddrescue_gui.Tools.DDRescueTools.allversions.get_initial_status(split_line)`

Function to get ddrescue's initial status.

**Args:**

**`split_line` (string):** The line from ddrescue's output that contains the information, split by whitespace.

Works with ddrescue versions: 1.14,1.15,1.16,1.17,1.18,1.19,1.20,1.21,1.22,1.23,1.24,1.25,1.26,1.27



DOCUMENTATION FOR THE DDRESCUE TOOLS FOR  
DDRESCUE VERSION 1.14 IN THE DDRESCUE TOOLS  
PACKAGE IN THE TOOLS PACKAGE  
(TOOLS/DDRESCUETOOLS/ONE\_POINT\_FORTTEEN.PY)

Tools for ddrescue v1.14 or newer.

`ddrescue_gui.Tools.DDRescueTools.one_point_fourteen.get_current_rate_error_size_recovered_data(split_line)`  
Get Current Read Rate, Error Size, and Recovered Data values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.14,1.15,1.16,1.17,1.18,1.19,1.20

`ddrescue_gui.Tools.DDRescueTools.one_point_fourteen.get_inputpos_numerrors_averagereadrate(split_line)`  
Get Input Position, Number of Errors, and Average Read Rate values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.14,1.15,1.16,1.17,1.18,1.19,1.20

`ddrescue_gui.Tools.DDRescueTools.one_point_fourteen.get_outputpos_time_since_last_read(split_line)`  
Get Output Position and Time Since Last Successful Read values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.14,1.15,1.16,1.17

`ddrescue_gui.Tools.DDRescueTools.one_point_fourteen.get_time_remaining(average_read_rate,  
average_read_rate_unit,  
disk_capacity,  
disk_capacity_unit,  
recovered_data)`

Calculate remaining time based on the average read rate and the current amount of data recovered.

**Returns:**

**`string.`** The remaining time in human-readable form eg "10.2 minutes", "4.3 days" etc, or  
"Unknown" if unable to calculate.

Works with ddrescue versions: 1.14,1.15,1.16,1.17,1.18,1.19

DOCUMENTATION FOR THE DDRESCUE TOOLS FOR  
DDRESCUE VERSION 1.18 IN THE DDRESCUE TOOLS  
PACKAGE IN THE TOOLS PACKAGE  
(TOOLS/DDRESCUETOOLS/ONE\_POINT\_EIGHTEEN.PY)

Tools for ddrescue v1.18 or newer.

`ddrescue_gui.Tools.DDRescueTools.one_point_eighteen.get_outputpos_time_since_last_read(split_line)`  
Get Output Position and Time Since Last Successful Read values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.18,1.19,1.20





DOCUMENTATION FOR THE DDRESCUE TOOLS FOR  
DDRESCUE VERSION 1.20 IN THE DDRESCUE TOOLS  
PACKAGE IN THE TOOLS PACKAGE  
(TOOLS/DDRESCUETOOLS/ONE\_POINT\_TWENTY.PY)

Tools for ddrescue v1.20 or newer.

`ddrescue_gui.Tools.DDRescueTools.one_point_twenty.get_time_remaining(split_line)`

Get Time Since Last Read value.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information, split by whitespace.

Works with ddrescue versions: 1.20,1.21,1.22,1.23,1.24,1.25,1.26,1.27

`ddrescue_gui.Tools.DDRescueTools.one_point_twenty.get_time_since_last_read(split_line)`

Get Time Since Last Read value.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information, split by whitespace.

Works with ddrescue versions: 1.20,1.21,1.22,1.23,1.24,1.25,1.26,1.27



DOCUMENTATION FOR THE DDRESCUE TOOLS FOR  
DDRESCUE VERSION 1.21  
IN THE DDRESCUE TOOLS PACKAGE IN THE TOOLS PACKAGE  
(TOOLS/DDRESCUETOOLS/ONE\_POINT\_TWENTY\_ONE.PY)

Tools for ddrescue v1.21 or newer.

`ddrescue_gui.Tools.DDRescueTools.one_point_twenty_one.get_current_rate_inputpos(split_line)`  
Get Current Read Rate and Input Position values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.21,1.22,1.23,1.24,1.25,1.26,1.27

`ddrescue_gui.Tools.DDRescueTools.one_point_twenty_one.get_outputpos_average_read_rate(split_line)`  
Get Output Position and Average Read Rate values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.21,1.22,1.23,1.24,1.25,1.26,1.27

`ddrescue_gui.Tools.DDRescueTools.one_point_twenty_one.get_recovered_data_num_errors(split_line)`  
Get Recovered Data and Number of Errors values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.21

`ddrescue_gui.Tools.DDRescueTools.one_point_twenty_one.get_unreadable_data(split_line)`  
Get Unreadable Data value.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.21,1.22,1.23,1.24,1.25,1.26,1.27



DOCUMENTATION FOR THE DDRESCUE TOOLS FOR  
DDRESCUE VERSION 1.22  
IN THE DDRESCUE TOOLS PACKAGE IN THE TOOLS PACKAGE  
(TOOLS/DDRESCUETOOLS/ONE\_POINT\_TWENTY\_TWO.PY)

Tools for ddrescue v1.22 or newer.

`ddrescue_gui.Tools.DDRescueTools.one_point_twenty_two.get_recovered_data_num_errors(split_line)`  
Get Recovered Data and Number of Errors values.

**Args:**

**`split_line (string):`** The line from ddrescue's output that contains the information,  
split by whitespace.

Works with ddrescue versions: 1.22,1.23,1.24,1.25,1.26,1.27



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`





## PYTHON MODULE INDEX

### d

- `ddrescue_gui.DDRescue_GUI`, 3
- `ddrescue_gui.tests`, 11
- `ddrescue_gui.Tests.__init__`, 13
- `ddrescue_gui.Tools`, 15
  - `ddrescue_gui.Tools.core`, 17
  - `ddrescue_gui.Tools.DDRescueTools`, 25
    - `ddrescue_gui.Tools.DDRescueTools.allversions`, 31
    - `ddrescue_gui.Tools.DDRescueTools.decorators`, 29
    - `ddrescue_gui.Tools.DDRescueTools.one_point_eighteen`, 35
    - `ddrescue_gui.Tools.DDRescueTools.one_point_fourteen`, 33
    - `ddrescue_gui.Tools.DDRescueTools.one_point_twenty`, 37
    - `ddrescue_gui.Tools.DDRescueTools.one_point_twenty_one`, 39
    - `ddrescue_gui.Tools.DDRescueTools.one_point_twenty_two`, 41
    - `ddrescue_gui.Tools.DDRescueTools.setup`, 27
  - `ddrescue_gui.Tools.mount_tools`, 21



## INDEX

### A

`add_line()` (*ddrescue\_gui.DDRescue\_GUI.CustomTextCtrl* method), 3

`attach_file()` (*ddrescue\_gui.Tools.mount\_tools.Mac* class method), 22

`AuthWindow` (class in *ddrescue\_gui.Tools.core*), 17

### B

`BackendThread` (class in *ddrescue\_gui.DDRescue\_GUI*), 3

`bind_events()` (*ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow* method), 4

`bind_events()` (*ddrescue\_gui.DDRescue\_GUI.FinishedWindow* method), 4

`bind_events()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5

`bind_events()` (*ddrescue\_gui.DDRescue\_GUI.PrivPolWindow* method), 9

`bind_events()` (*ddrescue\_gui.DDRescue\_GUI.SettingsWindow* method), 10

`bind_events()` (*ddrescue\_gui.Tools.core.AuthWindow* method), 17

`create_buttons()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5

`create_buttons()` (*ddrescue\_gui.DDRescue\_GUI.SettingsWindow* method), 10

`create_buttons()` (*ddrescue\_gui.Tools.core.AuthWindow* method), 17

`create_check_boxes()` (*ddrescue\_gui.DDRescue\_GUI.SettingsWindow* method), 10

`create_choice_boxes()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5

`create_choice_boxes()` (*ddrescue\_gui.DDRescue\_GUI.SettingsWindow* method), 10

`create_menus()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5

`create_other_widgets()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5

`create_other_widgets()` (*ddrescue\_gui.Tools.core.AuthWindow* method), 17

`create_text()` (*ddrescue\_gui.DDRescue\_GUI.FinishedWindow* method), 4

`create_text()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5

`create_text()` (*ddrescue\_gui.DDRescue\_GUI.SettingsWindow* method), 10

`create_text()` (*ddrescue\_gui.Tools.core.AuthWindow* method), 17

`create_unique_key()` (in module *ddrescue\_gui.Tools.core*), 18

### C

`carriage_return()` (*ddrescue\_gui.DDRescue\_GUI.CustomTextCtrl* method), 4

`change_units()` (in module *ddrescue\_gui.Tools.core*), 17

`check_for_updates()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5

`Core` (class in *ddrescue\_gui.Tools.mount\_tools*), 21

`create_buttons()` (*ddrescue\_gui.DDRescue\_GUI.FinishedWindow* method), 4

`create_widgets()` (*ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow* method), 4  
`create_widgets()` (*ddrescue\_gui.DDRescue\_GUI.PrivPolWindow* method), 9  
`CustomTextCtrl` (class in *ddrescue\_gui.DDRescue\_GUI*), 3

## D

`ddrescue_gui.DDRescue_GUI` module, 3  
`ddrescue_gui.tests` module, 11  
`ddrescue_gui.Tests.__init__` module, 13  
`ddrescue_gui.Tools` module, 15  
`ddrescue_gui.Tools.core` module, 17  
`ddrescue_gui.Tools.DDRescueTools` module, 25  
`ddrescue_gui.Tools.DDRescueTools.allversions` module, 31  
`ddrescue_gui.Tools.DDRescueTools.decorators` module, 29  
`ddrescue_gui.Tools.DDRescueTools.one_point_eighteen` module, 35  
`ddrescue_gui.Tools.DDRescueTools.one_point_fourteen` module, 33  
`ddrescue_gui.Tools.DDRescueTools.one_point_twenty` module, 37  
`ddrescue_gui.Tools.DDRescueTools.one_point_twenty_one` module, 39  
`ddrescue_gui.Tools.DDRescueTools.one_point_twenty_two` module, 41  
`ddrescue_gui.Tools.DDRescueTools.setup` module, 27  
`ddrescue_gui.Tools.mount_tools` module, 21  
`define_vars()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 5  
`define_versions()` (in module *ddrescue\_gui.Tools.DDRescueTools.decorators*), 29  
`determine_ddrescue_version()` (in module *ddrescue\_gui.Tools.core*), 18  
`determine_getdevinfo_version()` (in module *ddrescue\_gui.Tools.core*), 18  
`determine_output_file_type()` (*ddrescue\_gui.Tools.mount\_tools.Linux* class method), 21  
`determine_output_file_type()` (*ddrescue\_gui.Tools.mount\_tools.Mac* class method), 22  
`DiskInfoWindow` (class in *ddrescue\_gui.DDRescue\_GUI*), 4

## E

`ElapsedTimeThread` (class in *ddrescue\_gui.DDRescue\_GUI*), 4  
`emergency_exit()` (in module *ddrescue\_gui.Tools.core*), 18

## F

`file_choice_handler()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 6  
`find_ddrescue()` (in module *ddrescue\_gui.Tools.core*), 18  
`FinishedWindow` (class in *ddrescue\_gui.DDRescue\_GUI*), 4  
`focus_on_control_button()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 6

## G

`get_confirm_text()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 6  
`get_current_rate_error_size_recovered_data()` (in module *ddrescue\_gui.Tools.DDRescueTools.one\_point\_fourteen*), 33  
`get_current_rate_inputpos()` (in module *ddrescue\_gui.Tools.DDRescueTools.one\_point\_twenty\_one*), 39  
`get_device_name()` (*ddrescue\_gui.Tools.mount\_tools.Mac* class method), 22  
`get_diskinfo()` (*ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow* method), 4  
`get_diskinfo()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow* method), 6  
`get_helper()` (in module *ddrescue\_gui.Tools.core*), 18  
`get_info()` (*ddrescue\_gui.DDRescue\_GUI.GetDiskInformation* method), 5  
`get_initial_status()` (in module *ddrescue\_gui.Tools.DDRescueTools.allversions*), 31  
`get_inputpos_numerrors_averagereadrate()` (in module *ddrescue\_gui.Tools.DDRescueTools.allversions*), 31

```

        cue_gui.Tools.DDRescueTools.one_point_fourteen), method), 6
33 handle_outputfile_special_cases() (ddres-
get_mount_point() (in module ddres- cue_gui.DDRescue_GUI.MainWindow
        cue_gui.Tools.core), 19 method), 6
get_outputpos_average_read_rate() handle_user_file_selection() (ddres-
        (in module ddres- cue_gui.DDRescue_GUI.MainWindow
        cue_gui.Tools.DDRescueTools.one_point_twenty_one)method), 6
39
get_outputpos_time_since_last_read() |
        (in module ddres- is_mounted() (in module ddrescue_gui.Tools.core),
        cue_gui.Tools.DDRescueTools.one_point_eighteen), 19
35 is_partition() (in module ddres-
get_outputpos_time_since_last_read() cue_gui.Tools.core), 19
        (in module ddres-
        cue_gui.Tools.DDRescueTools.one_point_fourteen),
33 Linux (class in ddrescue_gui.Tools.mount_tools), 21
get_recovered_data_num_errors() M
        (in module ddres- Mac (class in ddrescue_gui.Tools.mount_tools), 22
        cue_gui.Tools.DDRescueTools.one_point_twenty_one),
39 MacReopenApp() (ddres-
get_recovered_data_num_errors() cue_gui.DDRescue_GUI.MyApp method),
        (in module ddres- 9
        cue_gui.Tools.DDRescueTools.one_point_twenty_two)
41 MainWindow (class in ddres-
        cue_gui.DDRescue_GUI), 5
get_time_remaining() (in module ddres- make_status_bar() (ddres-
        cue_gui.Tools.DDRescueTools.one_point_fourteen), cue_gui.DDRescue_GUI.MainWindow
        33 method), 6
get_time_remaining() (in module ddres- module
        cue_gui.Tools.DDRescueTools.one_point_twenty) ddrescue_gui.DDRescue_GUI, 3
37 ddrescue_gui.tests, 11
get_time_since_last_read() (in module ddres- ddrescue_gui.Tests.__init__, 13
        cue_gui.Tools.DDRescueTools.one_point_twenty) ddrescue_gui.Tools, 15
37 ddrescue_gui.Tools.core, 17
get_unreadable_data() (in module ddres- ddrescue_gui.Tools.DDRescueTools, 25
        cue_gui.Tools.DDRescueTools.one_point_twenty_one) ddrescue_gui.Tools.DDRescueTools.allversions,
39 31
get_volumes_apfs() (ddres- ddrescue_gui.Tools.DDRescueTools.decorators,
        cue_gui.Tools.mount_tools.Mac class 29
        method), 23 ddrescue_gui.Tools.DDRescueTools.one_point_eighteen,
get_volumes_lvm() (ddres- 35
        cue_gui.Tools.mount_tools.Linux class ddrescue_gui.Tools.DDRescueTools.one_point_fourteen,
        method), 21 33
get_volumes_std_device() (ddres- ddrescue_gui.Tools.DDRescueTools.one_point_twenty,
        cue_gui.Tools.mount_tools.Linux class 37
        method), 21 ddrescue_gui.Tools.DDRescueTools.one_point_twenty_one,
get_volumes_std_device() (ddres- 39
        cue_gui.Tools.mount_tools.Mac class ddrescue_gui.Tools.DDRescueTools.one_point_twenty_two,
        method), 23 41
GetDiskInformation (class in ddres- ddrescue_gui.Tools.DDRescueTools.setup,
        cue_gui.DDRescue_GUI), 5 27
H ddrescue_gui.Tools.mount_tools, 21
mount_device() (ddres-
handle_no_mapfile() (ddres- cue_gui.Tools.mount_tools.Linux class
        cue_gui.DDRescue_GUI.MainWindow method), 22

```

**mount\_device()** (ddrescue\_gui.Tools.mount\_tools.Mac class method), 23  
**mount\_disk()** (in module ddrescue\_gui.Tools.core), 19  
**mount\_output\_file()** (ddrescue\_gui.Tools.mount\_tools.Core class method), 21  
**mount\_partition()** (ddrescue\_gui.Tools.mount\_tools.Linux class method), 22  
**mount\_partition()** (ddrescue\_gui.Tools.mount\_tools.Mac class method), 23  
**MyApp** (class in ddrescue\_gui.DDRescue\_GUI), 9

**O**

**on\_abort()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 6  
**on\_about()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 6  
**on\_auth\_attempt()** (ddrescue\_gui.Tools.core.AuthWindow method), 17  
**on\_browse()** (ddrescue\_gui.DDRescue\_GUI.FinishedWindow method), 5  
**on\_close()** (ddrescue\_gui.DDRescue\_GUI.PrivPolWindow method), 9  
**on\_control\_button()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 6  
**on\_detailed\_info()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 6  
**on\_exit()** (ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow method), 4  
**on\_exit()** (ddrescue\_gui.DDRescue\_GUI.FinishedWindow method), 5  
**on\_exit()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**on\_exit()** (ddrescue\_gui.DDRescue\_GUI.ShowSplashWindow method), 10  
**on\_exit()** (ddrescue\_gui.Tools.core.AuthWindow method), 17  
**on\_mount()** (ddrescue\_gui.DDRescue\_GUI.FinishedWindow method), 5  
**on\_mount()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**on\_recovery\_ended()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**on\_session\_end()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7

**on\_size()** (ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow method), 4  
**on\_size()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**on\_start()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**on\_terminal\_output()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**OnInit()** (ddrescue\_gui.DDRescue\_GUI.MyApp method), 9

**P**

**PrivPolWindow** (class in ddrescue\_gui.DDRescue\_GUI), 9  
**process\_line()** (ddrescue\_gui.DDRescue\_GUI.BackendThread method), 3  
**prompt\_to\_kill\_ddrescue()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7

**R**

**read()** (in module ddrescue\_gui.Tools.core), 19  
**receive\_diskinfo()** (ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow method), 4  
**receive\_diskinfo()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**reset()** (ddrescue\_gui.Tools.mount\_tools.Core class method), 21  
**reset()** (ddrescue\_gui.Tools.mount\_tools.Linux class method), 22  
**reset()** (ddrescue\_gui.Tools.mount\_tools.Mac class method), 23  
**restart()** (ddrescue\_gui.DDRescue\_GUI.FinishedWindow method), 5  
**restart()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7  
**run()** (ddrescue\_gui.DDRescue\_GUI.BackendThread method), 3  
**run()** (ddrescue\_gui.DDRescue\_GUI.ElapsedTimeThread method), 4  
**run()** (ddrescue\_gui.DDRescue\_GUI.GetDiskInformationWindow method), 5  
**run()** (ddrescue\_gui.Tools.core.AuthWindow method), 17  
**run\_hdiutil()** (ddrescue\_gui.Tools.mount\_tools.Mac class method), 23

**S**

**save\_debug\_log()** (ddrescue\_gui.DDRescue\_GUI.MainWindow method), 7

`cue_gui.DDRescue_GUI.MainWindow`  
`method)`, 7

`save_options()` (`ddrescue_gui.DDRescue_GUI.SettingsWindow`  
`method)`, 10

`send_notification()` (*in module* `ddrescue_gui.Tools.core`), 20

`set_best_recovery_settings()` (`ddrescue_gui.DDRescue_GUI.SettingsWindow`  
`method)`, 10

`set_default_recovery_settings()` (`ddrescue_gui.DDRescue_GUI.SettingsWindow`  
`method)`, 10

`set_fast_recovery_settings()` (`ddrescue_gui.DDRescue_GUI.SettingsWindow`  
`method)`, 10

`set_input_file()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 7

`set_map_file()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 7

`set_output_file()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`set_progress_bar_range()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`set_soft_run()` (`ddrescue_gui.DDRescue_GUI.SettingsWindow`  
`method)`, 10

`set_vars()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`SettingsWindow` (*class in* `ddrescue_gui.DDRescue_GUI`), 10

`setup_for_ddrescue_version()` (*in module* `ddrescue_gui.Tools.DDRescueTools.setup`), 27

`setup_options()` (`ddrescue_gui.DDRescue_GUI.SettingsWindow`  
`method)`, 10

`setup_sizers()` (`ddrescue_gui.DDRescue_GUI.DiskInfoWindow`  
`method)`, 4

`setup_sizers()` (`ddrescue_gui.DDRescue_GUI.FinishedWindow`  
`method)`, 5

`setup_sizers()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`setup_sizers()` (`ddrescue_gui.DDRescue_GUI.PrivPolWindow`  
`method)`, 9

`setup_sizers()` (`ddrescue_gui.DDRescue_GUI.SettingsWindow`  
`method)`, 10

`setup_sizers()` (`ddrescue_gui.Tools.core.AuthWindow`  
`method)`, 17

`show_dev_info()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`show_inspector()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`show_privacy_policy()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`show_settings()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`show_userguide()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`ShowSplash` (*class in* `ddrescue_gui.DDRescue_GUI`), 10

`start_process()` (*in module* `ddrescue_gui.Tools.core`), 20

## T

`test_auth()` (`ddrescue_gui.Tools.core.AuthWindow`  
`method)`, 17

## U

`unmount_disk()` (*in module* `ddrescue_gui.Tools.core`), 20

`unmount_output_file()` (`ddrescue_gui.Tools.mount_tools.Core`  
`class method)`, 21

`unmount_output_file()` (`ddrescue_gui.Tools.mount_tools.Linux`  
`class method)`, 22

`unmount_output_file()` (`ddrescue_gui.Tools.mount_tools.Mac`  
`class method)`, 23

`up_one_line()` (`ddrescue_gui.DDRescue_GUI.CustomTextCtrl`  
`method)`, 4

`update()` (`ddrescue_gui.DDRescue_GUI.CustomTextCtrl`  
`method)`, 4

`update_average_read_rate()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`update_current_read_rate()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`update_error_size()` (`ddrescue_gui.DDRescue_GUI.MainWindow`  
`method)`, 8

`update_file_choices()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 8

`update_input_pos()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 8

`update_list_ctrl()` (*ddrescue\_gui.DDRescue\_GUI.DiskInfoWindow*  
*method*), 4

`update_num_errors()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 8

`update_output_pos()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 8

`update_progress()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 9

`update_recovered_data()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 9

`update_status_bar()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 9

`update_time_elapsed()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 9

`update_time_remaining()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 9

`update_time_since_last_read()` (*ddrescue\_gui.DDRescue\_GUI.MainWindow*  
*method*), 9

`usage()` (*in module ddrescue\_gui.DDRescue\_GUI*),  
10

`usage()` (*in module ddrescue\_gui.tests*), 11

X

`XYToPosition()` (*ddrescue\_gui.DDRescue\_GUI.CustomTextCtrl*  
*method*), 3