Davis Remmel

reMarkable Connection Utility

All-in-One Offline Tablet Management

User Manual

This is a really good license!

reMarkable Connection Utility User Manual
Version r2020.002

Contents

Do these page numbers carry into OCGs? &

Preface

You may have seen my work in the reMarkable tablet hacking community. I'm the one who added a microSD card to the tablet, then later installed a desktop GNU/Linux environment. My efforts focus on making the device usable in a general computing context.

This software, reMarkable Connection Utility (RCU), unshackles its users from reMarkable's proprietary cloud. The benefits are numerous: backups are full and fast, personal notes never leave the owner's control, users may personalize their device, and the author is actually responsive to the needs of the community. These are things the manufacturer won't provide.

RCU is *free*, not in price, but in liberty. Under the terms of its license, the GNU GPLv3, users hold the freedom to share this program with others, or even re-sell it. Anyone can make improvements because the source code is supplied with every purchase. This viral licensing forms a web of non-proprietary software, leading the world toward transparency and trust, precipitating software *rights*.

If you are a privacy-minded individual who wants to support independent software development that represents the needs of the tablet's community, please buy RCU. The funds generated will support me through writing a non-proprietary handwriting recognition engine, eventually authoring "magic paper" software influenced by Dynabook.

I would greatly appreciate your purchase; thank you.

Davis Remmel
Author of RCU

Like this

ii Support Information

This manual, and the RCU software, are available online from the official project page.

ii.1 General Support

Customers of RCU's original author are entitled to some email support. The author will try his best to satisfy each customer. Please write an email using the following header fields. Please reference the PayPal transaction ID in the message body.

To:	Davis Remmel <d@visr.me>
Subject:	RCU Support

ii.2 Getting Updates

Updates for RCU are announced via email to eligible customers; the program will not update automatically. People who buy RCU from the author will receive updates for one year. The program will never stop working; updates provide improvements, but a user can never be locked out of the software they own.

Recipients may unsubscribe from update announcements by sending an email to the author using the following header fields. Please reference the PayPal transaction ID in the message body.

To:	Davis Remmel <d@visr.me>
Subject:	Unsubscribe from RCU Update Announcements

ii.3 Bug Reporting

For advanced users who can identify a fault with the program, please submit a bug report via email with the following header fields. In the message body, please include: (a) a description of what is seen when using the program, (b) what is expected to be seen when using the program, (c) steps to reproduce the problem, and (d) information about the operating system and RCU version number (found in the About Pane).

To:	Davis Remmel <d@visr.me>
Subject:	RCU Bug: *Short description of problem*

1 Introduction

RCU allows complete offline management of a reMarkable tablet, without the need to connect with the manufacturer's proprietary cloud.

1.1 Compatibility

This software is tested with the following operationg systems.

- FreeBSD 12.1
- Ubuntu 18.04
- Apple macOS 10.13[1]
- Microsoft Windows 10[1]

[1]RCU is tested less-frequently in proprietary operating systems.

1.2 System Requirements

RCU will likely run on any computer manufactured after 2005. It requires approximately 100 megabytes of disk space, and may use up to 250 megabytes of memory.

1.3 Running RCU

RCU may connect to a tablet by USB or Wi-Fi. During periods of data transfer, never disconnect the tablet; doing so may result in a corrupted transfer. It is possible to upload a Recovery OS with RCU, which provides an emergency SSH connection over USB and allows the user to take or restore backups.

RCU is distributed as a single binary package. It does not need to be installed and will run from any directory. Running RCU is as easy as double-clicking on its executable icon.

The program will run with a console window open. This will print debug information. If something goes wrong, please let the author know, giving the information in this console window.

1.4 Entering Recovery Mode

The tablet may be placed into a recovery/flash mode with this sequence. If necessary, RCU can make and restore backups without there being a functional operating system. This mode is also necessary to install the Windows *libusb* driver.

1. Turn the device off.
2. Hold the middle facial button while turning the device on with the power button.
3. Continue holding the middle facial button for five seconds. The display will not update, but it is on.
4. The tablet should appear on the PC as *SE Blank MERGEZ*. It is now in recovery mode.

When done, restart the tablet by holding the power button for 10 seconds; release, then press the power button to turn it on normally.

1.5 Notes about GNU/Linux

In order to use the backup and restore functions in RCU, GNU/Linux hosts must grant read and write access to the tablet via *udev*. While in recovery mode, the tablet appears as a different USB device than normal operation.

Create a new udev ruleset at */etc/udev/rules.d/50-remarkable.rules*, as shown in Figure 1.1. Replace the *GROUP* attribute with a group belonging to the host's user. After creating this file, reboot the host computer.

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="15a2", ATTRS{idProduct}=="0061", \
  MODE="0660", GROUP="yourgroup"
SUBSYSTEM=="usb", ATTRS{idVendor}=="15a2", ATTRS{idProduct}=="0063", \
  MODE="0660", GROUP="yourgroup"
```

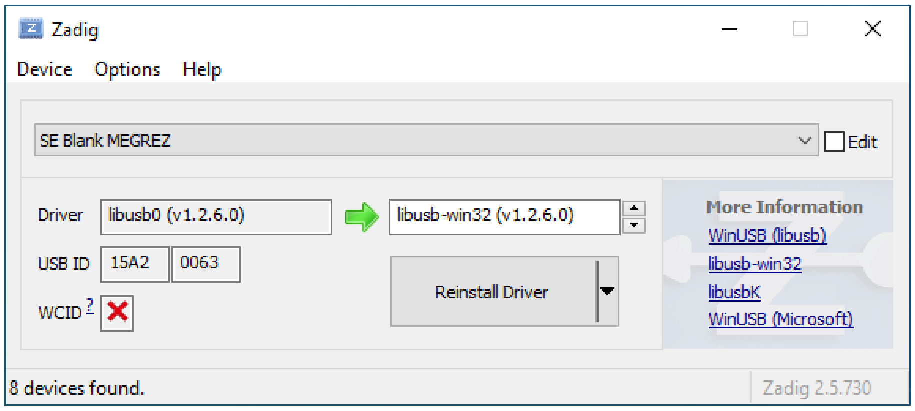
Figure 1.1: */etc/udev/rules.d/50-remarkable.rules*

1.6 Notes about Windows

RCU uses a USB interface library called *libusb*. In Windows, a driver must be installed to use the backup and restore features. Distributed with the RCU binary is a copy of Zadig, a utility that makes it simple to install this driver. First, the tablet must first be placed into recovery mode, which will appear as a new type of USB device.

Why does Microsoft make this so hard?

1. Connect the tablet to a PC with USB.
2. Put the tablet into recovery mode by following the steps in Entering Recovery Mode.
3. Open Zadig
   a) From the *Options* menu, enable *List All Devices*.
   b) In the device list, select *SE Blank MERGEZ*.
   c) Set the driver target to *libusb-win32*.
   d) Click *Install Driver* and wait for it to complete.
4. Hold the tablet's power button for 10 seconds; release, and press it again to turn the tablet on normally.
5. Reboot the PC



SE Blank MEGREZ

Driver libusb0 (v1.2.6.0) → libusb-win32 (v1.2.6.0)

USB ID 15A2 0063

WCID ? ✗

8 devices found. Zadig 2.5.730

Figure 1.2: Installing the *libusb* driver for Windows

2 Basic Operation

RCU is organized into separate panes, each handling a dedicated task. Panes may be switched by clicking on their titles in the left sidebar.

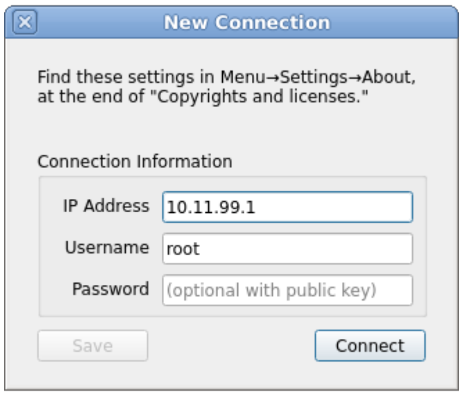
2.1 Connection Dialog

When RCU is launched, it will show the Connection Dialog. The user must enter the information used to connect to their tablet. These configuration settings may persist by clicking the Save button. Clicking the Connect button will initiate a connection to the device[1] and load the Panes window.

Figure 2.1 shows the Connection Dialog window.

[1] RCU uses SSH for all its communication with the tablet.

I like that it doesn't load any software



Find these settings in Menu→Settings→About, at the end of "Copyrights and licenses."

Connection Information

IP Address: 10.11.99.1
Username: root
Password: (optional with public key)

Figure 2.1: Connection Dialog

If a user finds themselves with an unresponsive tablet, they may place their device into a recovery mode by holding down the home button while pressing the power button. Right-click *Connect*, then click *Enter Recovery OS* (Figure 2.2).

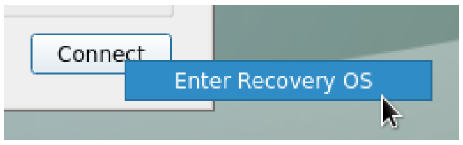


Figure 2.2: Enter Recovery OS

This is a very useful feature for development.

2.2 Device Info Pane

This pane shows the user basic information about their tablet, and allows the creation and restoring of tablet backups.

Rename

An owner may sign their name to their tablet using the Rename button. This only has one effect: changing the label from reading *Connected reMarkable* to *Name's reMarkable*.

Backups

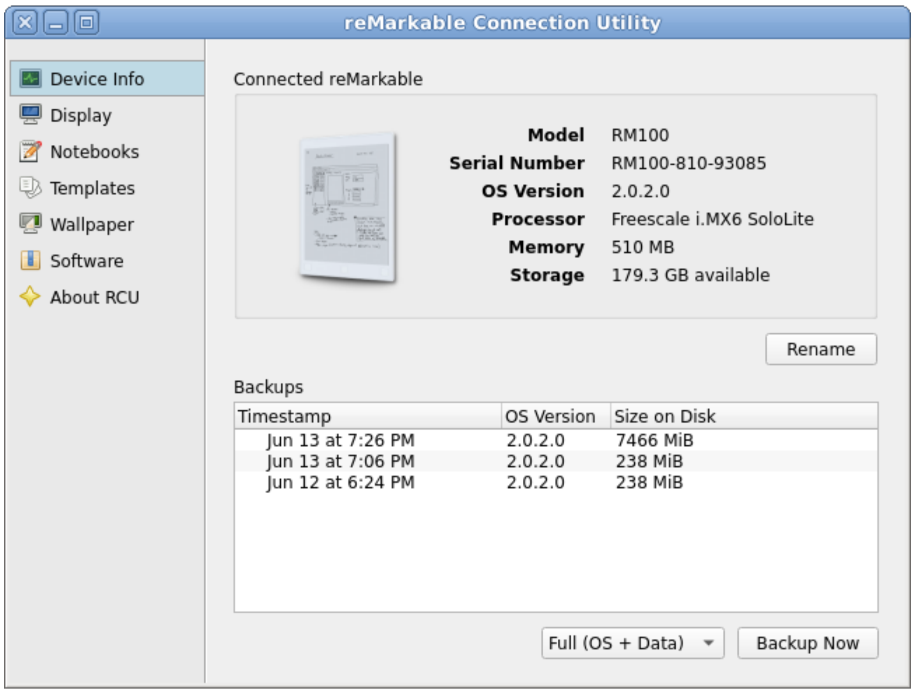
There are three backup types: OS, Data, and Full. Backups may only be taken or restored through a USB connection.

OS backups will restore the operating system partitions to the tablet's internal storage. They cannot be used to restore a bricked device. If a user applies an undesired OS update to their tablet, an OS backup may be used to downgrade to the prior OS without losing data. However, there can be no guarantee the data will remain compatible with the old OS image, such as when Xochitl was updated from version 1 to 2.

Data backups will restore the data partition, where documents reside, to the tablet's internal storage. They will not affect the operating system partitions, and are best used to revert a bulk of documents to an earlier state.

Full backups, as seen in Figure 2.4, may be used to restore a bricked device should the need ever arise. They require the most storage space on the client PC because they contain a complete mirror of the tablet's internal storage. A Full backup may be restored completely, or used to restore only the OS, or used to restore only the Data.

The backups created in this pane are block-level, not file-level, meaning they may only be restored to the device as they were taken. Although it is possible for advanced users to extract individual files from these backups, RCU cannot. If a user finds themselves in this situation, they may refer to the Backup Archive Format.



Connected reMarkable

Model	RM100
Serial Number	RM100-810-93085
OS Version	2.0.2.0
Processor	Freescale i.MX6 SoloLite
Memory	510 MB
Storage	179.3 GB available

Backups

Timestamp	OS Version	Size on Disk
Jun 13 at 7:26 PM	2.0.2.0	7466 MiB
Jun 13 at 7:06 PM	2.0.2.0	238 MiB
Jun 12 at 6:24 PM	2.0.2.0	238 MiB

Figure 2.3: Device Info Pane

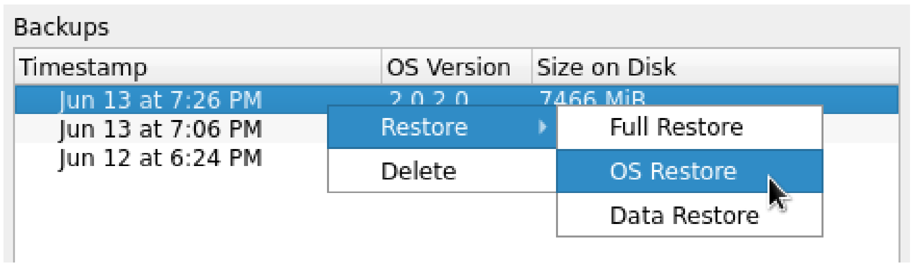


Figure 2.4: A Full backup may be restored in-full, or with only OS or Data partitions.

2.3 Display Pane

A user may capture screenshots of their tablet through the Display Pane. Press the Refresh button to preview the screen, then press the Save Screenshot button to record the image[2] to disk.

[2] Although the tablet's framebuffer encodes the image as 16-bit RGB (RGB565), it is saved to disk as an 8-bit grayscale PNG.

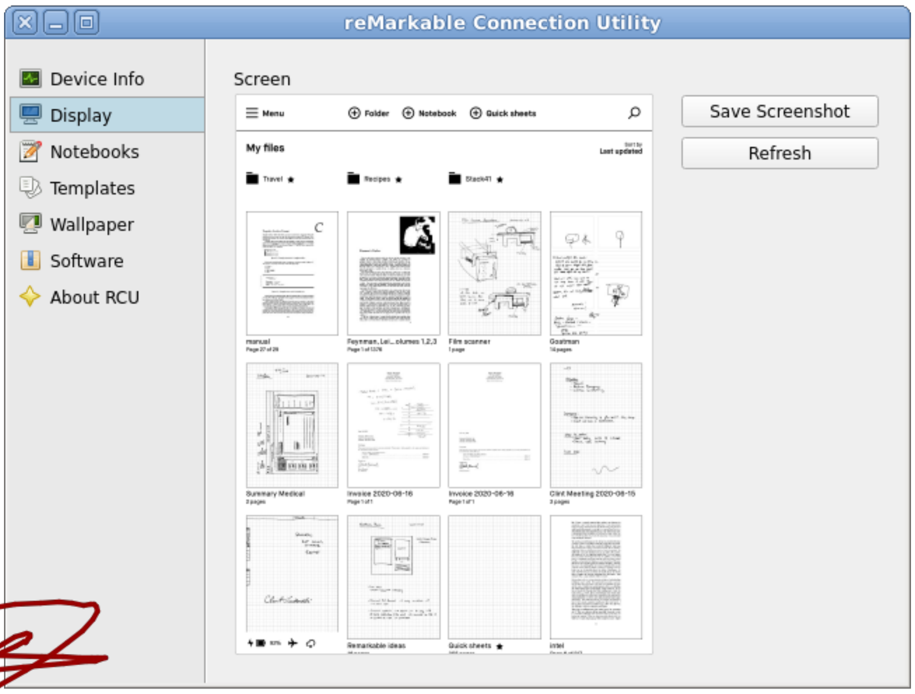
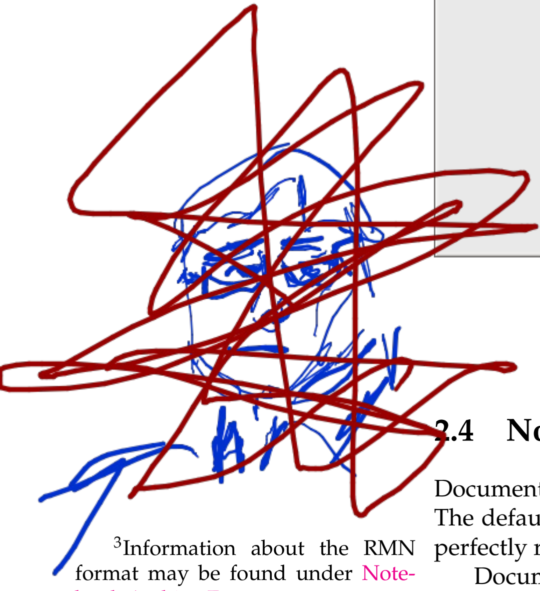


Figure 2.5: Display Pane



2.4 Notebooks Pane

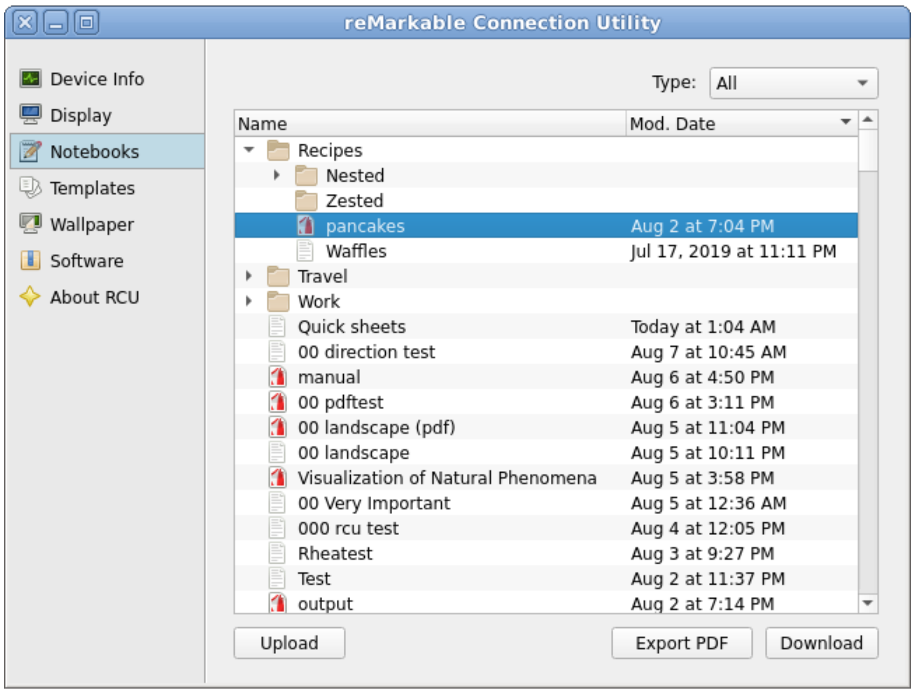
Documents may be transmitted between a tablet and PC through the Notebooks Pane. The default download type is a reMarkable Notebook Archive (RMN) because it can perfectly restore notebooks and their templates[3].

[3] Information about the RMN format may be found under Notebook Archive Format.

Documents may be uploaded to the device as RMN or PDF files. Click the Upload button to select the file(s) to upload.

Documents may be downloaded from the device as RMN or PDF files. It is recommended to archive notebooks with the RMN format because they are lossless, containing all the information needed to recreate a PDF. When downloading a single file, it may be renamed before it is saved. When downloading files as a batch the user must select a directory to save them in.

Deleting documents is performed in the tree view through a contextual menu. Right-click on an item, then click Delete. The user is prompted for confirmation, then the document is permanantly deleted.



Type: All

Name	Mod. Date
Recipes	
Nested	
Zested	
pancakes	Aug 2 at 7:04 PM
Waffles	Jul 17, 2019 at 11:11 PM
Travel	
Work	
Quick sheets	Today at 1:04 AM
00 direction test	Aug 7 at 10:45 AM
manual	Aug 6 at 4:50 PM
00 pdftest	Aug 6 at 3:11 PM
00 landscape (pdf)	Aug 5 at 11:04 PM
00 landscape	Aug 5 at 10:11 PM
Visualization of Natural Phenomena	Aug 5 at 3:58 PM
00 Very Important	Aug 5 at 12:36 AM
000 rcu test	Aug 4 at 12:05 PM
Rheatest	Aug 3 at 9:27 PM
Test	Aug 2 at 11:37 PM
output	Aug 2 at 7:14 PM

Figure 2.6: Notebooks Pane

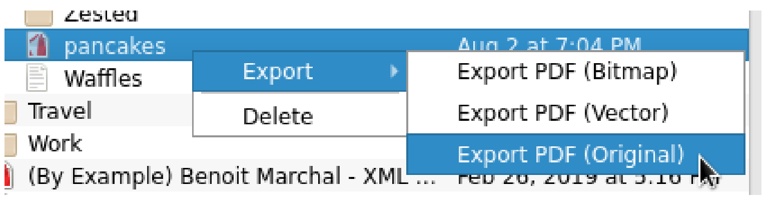


Figure 2.7: Options when exporting a PDF document

2.5 Templates Pane

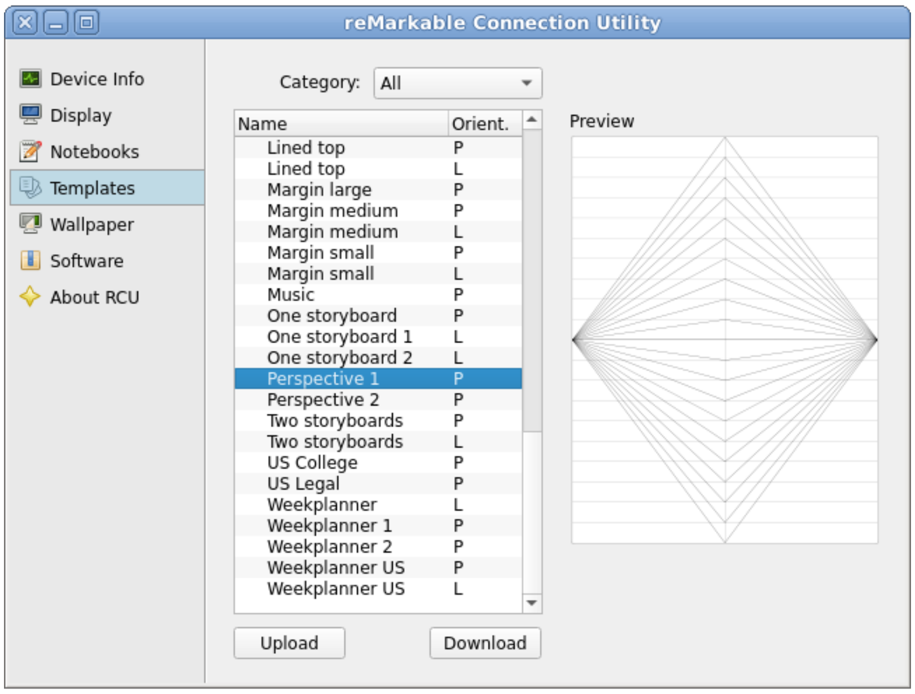
Users may add or remove their own templates in the reMarkable Template Archive (RMT) format. RCU will not accept templates in PNG or other image-based formats. For information about how to make an RMT, please read the Template Archive Format. It is easy, being a tape archive (.tar) of an SVG and a JSON file.

Add a template to the tablet by clicking the Upload button, then selecting an RMT file.

Download a template from the tablet by selecting one in the list view, clicking the Download button, then choosing a filename to save.

Delete a template from the tablet through a contextual menu by right-clicking a template in the list view, then clicking Delete. After confirmation, RCU will permanantly delete the template from the device.

Template are installed to the user's home directory, in */.local/share/remarkable/templates*. Softlinks are created where the system templates are stored, in */usr/share/remarkable/templates*. A system update may remove these links, and the templates will not load in the tablet's interface. If the user previously installed custom templates using RCU, this situation will be detected, the program will alert the user, and the template links may be restored automatically.



Category: All

Name	Orient.
Lined top	P
Lined top	L
Margin large	P
Margin medium	P
Margin medium	L
Margin small	P
Margin small	L
Music	P
One storyboard	P
One storyboard 1	L
One storyboard 2	L
Perspective 1	P
Perspective 2	P
Two storyboards	P
Two storyboards	L
US College	P
US Legal	P
Weekplanner	L
Weekplanner 1	P
Weekplanner 2	P
Weekplanner US	P
Weekplanner US	L

Figure 2.8: Templates Pane

2.6 Wallpaper Pane

Device wallpapers[4] may be changed for the Suspend and Poweroff screens. It is recommended these images have a resolution of 1404×1872 pixels, and must be in the PNG format.

The Suspend screen displays when a user presses the top (physical) button of the device. Users may update this wallpaper by clicking the Upload button (beneath the Suspend label), then selecting a PNG image.

The Poweroff screen displays when a user holds the top (physical) button of the device. Users may update this wallpaper by clicking the Upload button (beneath the Poweroff label), then selecting a PNG image.

Images may be reset to the factory-defaults by clicking the Set To Original button.

[4]These are sometimes called *splash images*.

"Wallpaper" has a better ring

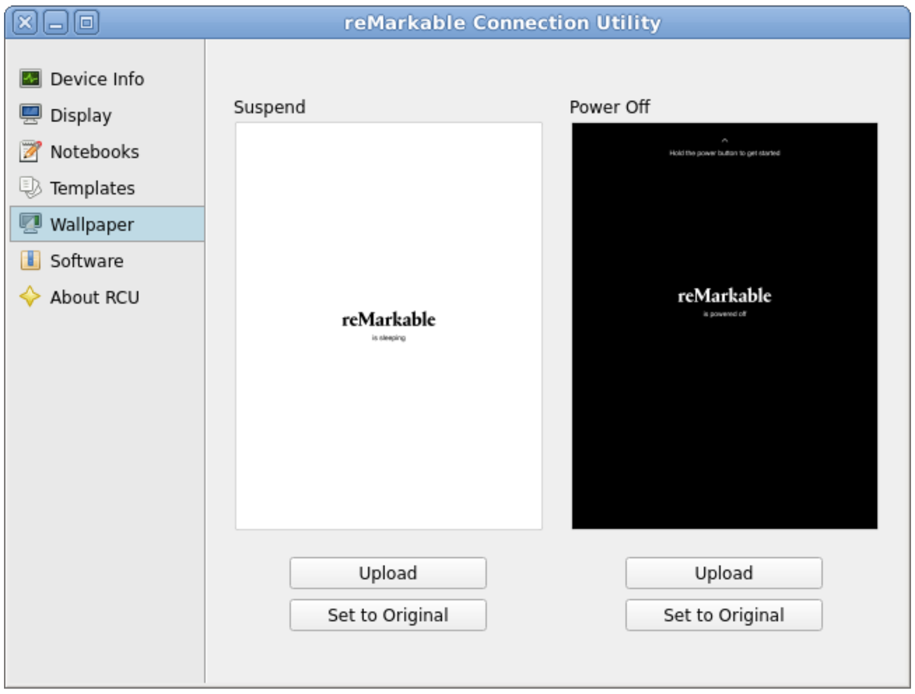


Figure 2.9: Wallpaper Pane

2.7 Software Pane

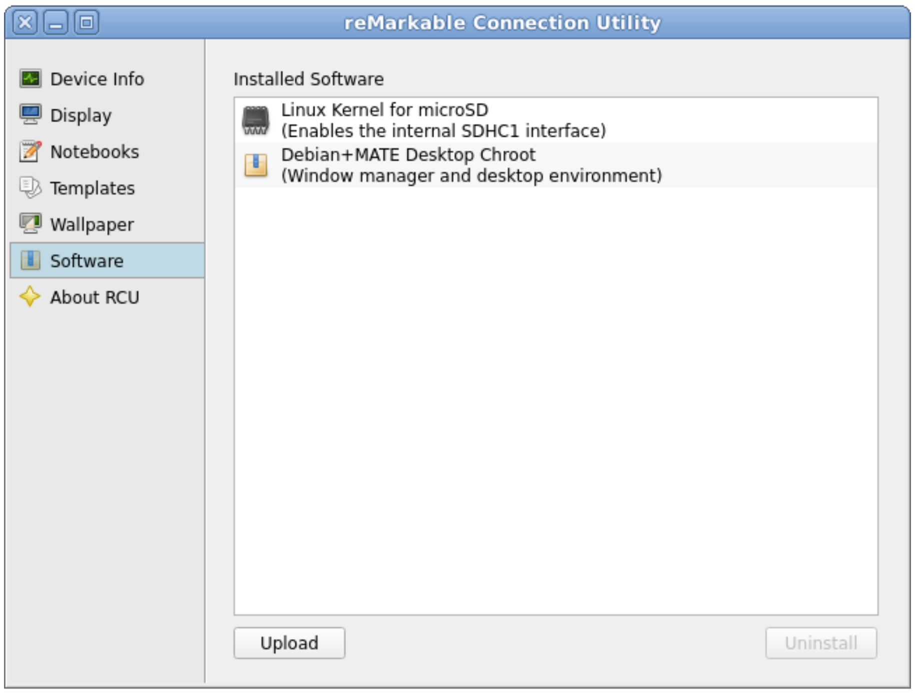
Third-party software may be uploaded to the device in the reMarkable Software Package (RMPKG) format[5].

[5] For details about creating an RMPKG file, please read the Software Package Format.

To install a software package, click the Upload button, select an RMPKG file, then wait for the install process to complete.

To remove a software package, select one in the list view, then click the Uninstall button and wait for the removal process to complete.

Neat.



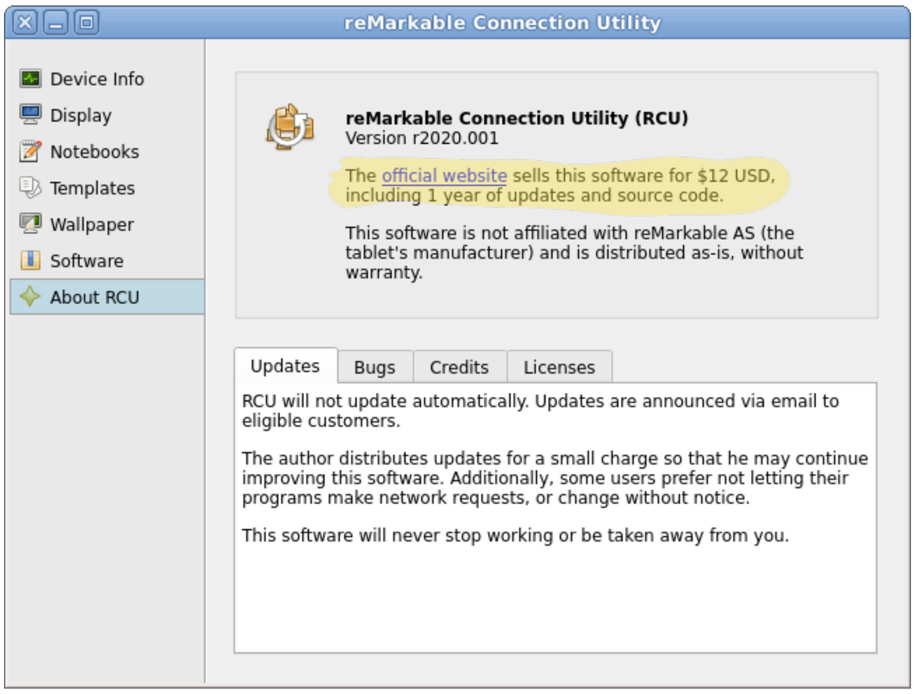
Linux Kernel for microSD
(Enables the internal SDHC1 interface)
Debian+MATE Desktop Chroot
(Window manager and desktop environment)

Figure 2.10: Software Pane

2.8 About Pane

Information about RCU may be viewed in the About pane. The information panel contains its version number. Credits to the software used to build RCU, and copies of their licenses, are also available.

By clicking the package icon image, a user may request RCU to contact the update server to check if they are running the latest version.



reMarkable Connection Utility (RCU)
Version r2020.001

The official website sells this software for $12 USD, including 1 year of updates and source code.

This software is not affiliated with reMarkable AS (the tablet's manufacturer) and is distributed as-is, without warranty.

RCU will not update automatically. Updates are announced via email to eligible customers.

The author distributes updates for a small charge so that he may continue improving this software. Additionally, some users prefer not letting their programs make network requests, or change without notice.

This software will never stop working or be taken away from you.

Figure 2.11: About Pane

3 Developing with RCU

3.1 Running from Source

RCU requires a few pieces of software, listed below, when running directly from its source code.

- Python 3.6+
- Qt 5.14+[1]

[1] FreeBSD-only, since PySide2 is not available through *pip*

Furthermore, a few Python dependencies must be installed, which are specified in *src/requirements.txt*. It is recommended to install dependencies within a *venv* environment.

- PySide2 5.14+
- Paramiko 2.7+
- pdfrw 0.4+
- PyInstaller 3.6

Once these software requirements are satisfied, execute *src/main.py* with the Python interpreter.

3.2 Building a Release Binary

For official releases, RCU's source is amalgamated with various other utilities into a single executable binary with PyInstaller. For convenience, this process has been automated as a GNU makefile.

To build a release image, execute *make* from a terminal. The resulting file will be an archive that includes a Python interpreter and all dependant libraries. Upon execution, this binary deflates to, and starts running from, a temporary directory.

To build the documentation, execute *make doc*. This will build the manual in PDF format[2].

[2] Building the documentation requires *pdflatex*, which is not covered here.

3.3 Adding Custom Panes

The Pane Architecture of RCU is modular, so new panes may be added easily. To write a new pane, first create a new directory under *src/panes*, where the pane's source code will be stored. In that directory, create a file, *pane.py*, which will serve as the focal point of execution.

The *pane.py* file must contain a class for the pane, implementing the following requirements: (a) the pane must inherit from the *UIController* class[3], (b) it must provide the *name* and *ui_filename* class variables, and (c) it must initialize first through the parent class. An example is listed in Figure 3.1.

[3]Find the UIController source code at *src/controllers/UIController.py*.

The pane must be accompanied by a Qt UI file, specified in the class variable *ExamplePane.ui_filename*. This UI file should be a widget with dimensions of 440×480 pixels, and not be re-sizable. This is exposed to the pane's class as *self.window* upon instantiation.

After adding the *pane.py* file, import it within *src/panes/__init__.py* (Figure 3.2). Once added, the pane will draw itself into RCU's main window.

For non-immediate tasks, it is recommended to use a *worker*. RCU's GUI runs on the main thread, and blocking this may provide a poor user experience. Workers may be executed in the thread pool, which return to the main thread via asynchronous callback. For examples of worker usage, please read the source code for one of the default panes.

```
'''
pane.py
This is an example pane.

License: GPLv3 or later
'''

import log
from Worker import worker
from pathlib import Path
from controllers import UIController

class ExamplePane(UIController):
    name = 'Example Pane'

    # Dynamic path loading works when running from source
    # and binary.
    adir = Path(__file__).parent.parent
    bdir = Path(__file__).parent
    ui_filename = Path(adir / bdir / 'example.ui')

    xochitl_versions = [
        '^2\.2\.0\.[0-9]+$'
    ]

    @classmethod
    def get_icon(cls):
        ipathstr = str(Path(cls.bdir / 'icons' / 'emblem-documents.png'))
        icon = QIcon()
        icon.addFile(ipathstr, QSize(16, 16), QIcon.Normal, QIcon.On)
        return icon

    def __init__(self, pane_controller):
        super(type(self), self).__init__(
            pane_controller.model, pane_controller.threadpool)
        # Exposed now are self.model, self.window, and
        # self.threadpool.
        # ...
```

This guy doesn't hide anything!

Figure 3.1: Example source code for *pane.py*

```
from .example.pane import ExamplePane

paneslist = [
    # ...
    ExamplePane
    ]
```

Figure 3.2: Example of importing *pane.py* in *src/panes/__init__.py*

A

Backup Archive Format

Backups are stored on disk next to RCU's preferences. This path varies depending on the operating system. Refer to the Locations Where Application Settings Are Stored section, in the QSettings documentation, to determine the specific system path. In FreeBSD backups are stored in *~/.config/davisr/rcu/backups*.

Each backup archive is given a unique identifier, and a directory is created to store its contents. An example directory structure is listed in Figure A.1

```
backups
└─ 902b512b-8742-481d-b5f1-e185c0668e9f
   ├─ files
   │  ├─ mmcblk1boot0.bin
   │  ├─ mmcblk1boot1.bin
   │  └─ mmcblk1.bin
   └─ backup.json
```

Figure A.1: Example structure of a backup archive

The *backup.json* file contains metadata about the backup, and is used by RCU to populate the UI. In summary, this file contains the backup's ID, timestamp, device information, the device's partition table (the output of *fdisk -l*), and checksums of the dumped partitions.

OS backups store the *u-boot* bootloader, secondary boot partition (containing factory device information), the bootloader data partition, primary OS partition, and secondary OS partition. The primary OS may reside on *mmcblk1p2* or *mmcblk1p3*, flipping after every system update.

- /dev/mmcblk1boot0
- /dev/mmcblk1boot1
- /dev/mmcblk1p1
- /dev/mmcblk1p2
- /dev/mmcblk1p3

Data backups only store the data partition, which is mounted as */home/root*.

- /dev/mmcblk1p7

Full backups store the *u-boot* bootloader, secondary boot partition, and the entire contents of the eMMC (all partitions combined).

- /dev/mmcblk1boot0
- /dev/mmcblk1boot1
- /dev/mmcblk1

B Notebook Archive Format

Notebook Archive (RMN) files contain the raw data files used by Xochitl, and a copy of all template used by that document in the Template Archive Format. They have an obvious structure, as seen in Figure B.1. This directory is a direct export from Xochitl's files, from the device at *~/.local/share/remarkable/xochitl*.

```
ExampleNotebook.rmn
├── 6bde82bd-f580-456b-8275-c853438707a6
│   ├── 5ae652c8-280b-4b00-9563-72f25f16ac29.rm
│   ├── 3e9610c9-7633-4964-8198-adc0d1968cea.rm
│   └── (...)
├── 6bde82bd-f580-456b-8275-c853438707a6.content
├── 6bde82bd-f580-456b-8275-c853438707a6.metadata
├── 6bde82bd-f580-456b-8275-c853438707a6.pagedata
├── Blank.rmt
└── Small Lines.rmt
```

Figure B.1: Example structure of a template archive

When saving a document it is preferred to use the RMN format over the PDF format because RMN files contain an exact copy of those notebooks. Therefore, a PDF may always be generated from an RMN archive.

C Template Archive Format

Template Archive (RMT) files follow an easy-to-create format. Typically, reMarkable templates have been shared amongst the community as singular PNG images. However, this has a number of drawbacks, like being stuck with a static resolution and lack of metadata.

The RMT format solves these issue by bundling a vector image of the template, instead of a bitmap image, and includes template metadata in the archive. An example file structure of an RMT file may be seen in Figure C.1. The RMT file is a tape archive (TAR) file with the *.rmt* extension.

```
ExampleTemplate.rmt
 └─template.json
 └─template.svg
```

Figure C.1: Example structure of a template archive

The *template.json* file should contain, at minimum, a structure similar to Figure C.2. The *categories* array may contain any of the following strings.

- Creative
- Grids
- Life/organize
- Lines

```json
{
    "categories": [
        "Creative"
    ],
    "iconCode": "\ue9d5",
    "landscape": false,
    "name": "Example Template"
}
```

Figure C.2: Example source code for *template.json*

The *template.svg* file should contain valid SVG data and have a viewport resolution of 1404×1872 pixels. For viewability with an e-paper screen, it is recommended not to have features smaller than 2 pixels.

When a template archive is uploaded, these files are not extracted to the default template location (*/usr/share/remarkable/templates*). Instead, they are extracted into the home directory, at *˜/.local/share/remarkable/templates*. Upon upload, RCU will convert the SVG image to a PNG image to retain compatibility with Xochitl.

If the tablet receives an update that clears the system template directory, the templates will become unavailable from the interface. RCU may detect this condition and prompt the user to recreate these template links, in-effect restoring the templates (they do not need to be re-uploaded).

D Software Package Format

Please download the RCU sources, then find the *rmpkg-sample* directory. This sample package contains lots of useful information about the package format in a practical way.

The RMPKG format is a self-contained executable. The easiest way to make this is with an ordinary shell script, appended with the application's binary payload (like a TAR). The package should expose the following flags.

- *--info*: Prints information about the package useful for anyone who stumbles upon it
- *--manifest*: Prints a manifest of any files touched by the package; used for detecting conflicting packages
- *--install*: Installs the package payload to the system
- *--uninstall*: Uninstalls the package payload from the system