

# RUT955 USB Tools (legacy WebUI)

[Main Page](#) > [RUT Routers](#) > [RUT955](#) > [RUT955 Manual](#) > [RUT955 Legacy WebUI](#) > [RUT955 Services section \(legacy\)](#) > **RUT955 USB Tools (legacy WebUI)**

The information in this page is updated in accordance with firmware version [RUT9XX\\_R\\_00.06.09.5](#).

**Note:** this user manual page is for RUT955's old WebUI style available in earlier FW versions. [Click here](#) for information based on the latest FW version.

□

## Contents

- [1 Summary](#)
- [2 Network Shares](#)
  - [2.1 Mounted File Systems](#)
    - [2.1.1 SSH commands](#)
      - [2.1.1.1 Mounting](#)
      - [2.1.1.2 Unmounting](#)
  - [2.2 Samba](#)
    - [2.2.1 Shared Directories](#)
  - [2.3 Samba Users](#)
- [3 Printer Server](#)
- [4 USB to Serial](#)
- [5 USB Modem](#)

## Summary

The **USB Tools** page is used to manage services related to the device's USB connector.

This chapter of the user manual provides an overview of the USB Tools page in RUT955 devices.

---

USB Tools is additional software that can be installed from the **System** → [Package Manager](#) page.

## Network Shares

The **Network Shares** section is used to manage Network-attached storage (NAS) such as USB drives, SD cards and hard drives. The router supports the following file system architectures:

- **FAT32\***
- **NTFS\*\***

\* supports SD cards with a capacity of up to **32 GB**

\*\* NTFS support available from FW version **RUT9XX\_R\_00.05.02** and newer

## Mounted File Systems

---

The **Mounted File Systems** tab displays information on mounted USB drives and SD cards. To view an attached drive, connect it to the router and click the Refresh button:



field name	description
Filesystem	Name of the Linux filesystem. For better clarity, lets take a closer look at the filesystem name from the example above: <ul style="list-style-type: none"><li>• <b>/dev</b> - the directory of device files. Everything in the Linux filesystem is either a file or a directory; /dev is the location of device files. Every device is represented by a file in the /dev directory</li><li>• <b>sd</b> - identifies block devices* (capable of carrying data)</li><li>• <b>a</b> - the letter immediately after sd signifies the order in which the device was first found (a, b, c, ...)</li><li>• <b>1</b> - signifies the partition on the device</li></ul> So the full string <b>/dev/sda1</b> denotes the first partition of the first block device <ul style="list-style-type: none"><li>* SD cards will be identified as MultiMediaCard (MMC) devices and the identifier will begin with the <b>mmcblk</b> string instead of sd</li></ul>
Mount Point	A mount point is a directory in the currently accessible filesystem on which an additional filesystem is mounted (i.e., logically attached). The <b>/mnt</b> directory and its subdirectories are intended for use as the temporary mount points for mounting storage devices. The mount point becomes the root directory of the newly added filesystem, and that filesystem becomes accessible from that directory
Available	Amount of available memory in the mounted drive. This column also displays the total capacity of the drive
Used	Amount of used memory in the mounted drive
Safely Remove Disk	Unmounts the drive so that it becomes safe to disconnect from the router

## SSH commands

---

Devices can be mounted or unmounted manually by using SSH commands.

### Mounting

---

The **mount** command attaches a filesystem to the file hierarchy. The basic syntax for the command is:

```
mount -t type device dir
```

Where:

- **-t** - the filesystem type. If this value is omitted, the command will try to determine the type automatically
- **device** - file name of the device. Refer to section [2.1](#) for examples
- **dir** - directory that will be used as the mount point for the device. Refer to section [2.1](#) for examples

### Example command:

```
mount -t ntfs /dev/sda1 /mnt/sda1
```

### Unmounting

---

The **umount** command detaches filesystem from the file hierarchy. The basic syntax for the command is:

```
umount device
```

## Example command:

```
umount /dev/sda1
```

Using the umount command is analogous to clicking the "Safely Remove Disk" button. Therefore, the attached drive can be safely disconnected after using umount to detach it.

## Samba

---

**Samba** is a software solution for using the Server Message Block (SMB) networking protocol, which provides shared file access between nodes on a computer network. The Samba tab in RUT955's WebUI is used to configure network access (look to section [2.2.1](#) for details) to storage devices connected to the router's MicroSD slot or USB connector.



Field	Value	Description
Enable	yes   no; default: <b>no</b>	Turns Samba on or off.
Hostname	string; default: <b>Router_share</b>	Name of the Samba server.
Description	string; default: <b>Router_share</b>	Short description about the Same server.
Workgroup	string; default: <b>WORKGROUP</b>	Name of the server's workgroup.

## Shared Directories

---

The **Shared Directories** section is used to configure access to connected storage drives or individual directories contained in those drives. A shared directory must be added before it can be configured:



Field	Value	Description
Name	string; default: <b>none</b>	Name of a shared directory.
Path	filepath; default: <b>/mnt/</b>	Path to a shared directory. To share an entire drive, choose an automatically generated path from this drop-down box (for example, /mnt/sda1). To share a specific directory on the drive, specify the full path to that directory (for example, /mnt/sda1/shared/video).
Allow guests	yes   no; default: <b>no</b>	Turns guest access on or off. Guest access allows anonymous connections to a Shared Directory.
Allowed users	samba user(s); default: <b>none</b>	Samba user(s) that are permitted to access a Shared Directory. Users can be created from the Samba Users section.
Read-only	yes   no; default: <b>no</b>	Makes a Shared Directory read-only, which means the directory can only be accessed to view and read files (not write).
Delete	-(interactive button)	Deletes a Shared Directory configuration.

## Samba Users

---

**Samba users** are users that can be granted access to shared directories (as described in section [2.2.1](#)). To add a new user, enter a custom username and password and click the 'Add' button:



Field	Value	Description
Username	string; default: <b>none</b>	Custom name for a new Samba user.
Password	string; default: <b>none</b>	Password for a new Samba user.

## Printer Server

The **Printer Server** provides the possibility to configure access to a printer that is connected to the USB port of the RUT955 device. The Add button lets you add and manage additional printers. To configure a printer instance, click the Edit button located next to it:



Field	Value	Description
Enable	yes   no; default: <b>no</b>	Turns USB printer support on or off.
Device	filepath; default: <b>/dev/usb/lp0</b>	Printer's device file.
Port	integer [9100..9109]; default: <b>9100</b>	Printer's TCP port.
Bidirectional mode	off   on; default: <b>on</b>	Turns bidirectional mode on or off.

For step-by-step instructions on how to use a printer with RUT955, [click here](#)

## USB to Serial

The router's USB connector can also be used as a serial port. The only type of serial connection available via USB is Over IP.

In **Over IP** serial type the device provides a connection to a TCP/IP network for the devices connected via the serial interface. The figure below is an example of available configuration fields for the Over IP serial type and the table below provides information on these fields:



Field	Value	Description
Enabled	yes   no; default: <b>no</b>	Turns the USB to Serial service on or off.
Baud rate	300   1200   2400   4800   9600   19200   38400   57600   115200; default: <b>115200</b>	Data rate for serial data transmission (in bits per second).
Data bits	5   6   7   8; default: <b>8</b>	Number of data bits for each character

Parity	None   Odd   Even; default: <b>None</b>	<p>In serial transmission, parity is a method of detecting errors. An extra data bit is sent with each data character, arranged so that the number of 1 bits in each character, including the parity bit, is always odd or always even. If a byte is received with the wrong number of 1s, then it must have been corrupted. However, an even number of errors can pass the parity check.</p> <ul style="list-style-type: none"> <li>• <b>None (N)</b> - no parity method is used</li> <li>• <b>Odd (O)</b> - the parity bit is set so that the number of "logical ones (1s)" has to be odd.</li> <li>• <b>Even (E)</b> - the parity bit is set so that the number of "logical ones (1s)" has to be even.</li> </ul>
Stop bits	1   2; default: <b>1</b>	<p>Stop bits sent at the end of every character allow the receiving signal hardware to detect the end of a character and to resynchronise with the character stream. Electronic devices usually use one stop bit. Two stop bits are required if slow electromechanical devices are used.</p>
Flow control	None   RTS/CTS   Xon/Xoff; default: <b>None</b>	<p>In many circumstances a transmitter might be able to send data faster than the receiver is able to process it. To cope with this, serial lines often incorporate a "handshaking" method, usually distinguished between hardware and software handshaking.</p> <ul style="list-style-type: none"> <li>• <b>RTS/CTS</b> - hardware handshaking. RTS and CTS are turned OFF and ON from alternate ends to control data flow, for instance when a buffer is almost full.</li> <li>• <b>Xon/Xoff</b> - software handshaking. The Xon and Xoff characters are sent by the receiver to the sender to control when the sender will send data, i.e., these characters go in the opposite direction to the data being sent. The circuit starts in the "sending allowed" state. When the receiver's buffers approach capacity, the receiver sends the Xoff character to tell the sender to stop sending data. Later, after the receiver has emptied its buffers, it sends an Xon character to tell the sender to resume transmission.</li> </ul>
Serial type	Over IP; default: <b>Over IP</b>	Specifies the serial connection type.
Protocol	TCP; default: <b>TCP</b>	Protocol used in the communication process
Mode	Server   Client   Bidirect; default: <b>Server</b>	<p>The role of this device in the serial connection:</p> <ul style="list-style-type: none"> <li>• <b>Server</b> - the device waits for incoming connections</li> <li>• <b>Client</b> - the device initiates the connection.</li> <li>• <b>Bidirect</b> - acts as client by default but waits for incoming connections at the same time.</li> </ul>
No leading zeros	yes   no; default: <b>no</b>	When checked, indicates that the first hex zeros should be skipped
Client settings: Server address	ip   host; default: <b>none</b>	IP address or hostname of the server that the client will connect to
Client settings: TCP port	integer [0..65535]; default: <b>none</b>	Server's port number
Client settings: Reconnect interval (s)	integer; default: <b>none</b>	Time period (in seconds) between reconnection attempts in case a connection fails
Server settings: TCP port	integer [0..65535]; default: <b>none</b>	Internal TCP port number used to listen for incoming connections

<b>Server settings:</b>	integer; default:	Disconnects clients after they remain inactive for an amount
Timeout (s)	<b>none</b>	time (in seconds) specified in this field
<b>Output</b>	OC Output   Relay Output; default: <b>OC</b> <b>Output</b>	Output to indicate that the application switched from client (default) to server state (this field becomes visible only in Bidirect mode)
<b>Output state</b>	1   0; default: <b>0</b>	Output state value during which the application reverts to server mode (this field becomes visible only in Bidirect mode)

## USB Modem

[Click here](#) for step-by-step USB modem set up instructions.

RUT955 devices are compatible with the following USB modems:

- Huawei MS2131i-8