Depending on how much information is provided, either the values specified in the parameter or the values specified with the command `MMEM:CDIR` (default directory) are used for the path and the drive settings in the commands.

Before the instrument settings can be stored in a file, they have to be stored in an intermediate memory using common command `*SAV <number>`. The specified number is subsequently used in the `:MMEMory:STORe:STATe` on page 313 command. Also, subsequently to loading a file with instrument settings with command `:MMEMory:LOAD:STATe` on page 312, these settings have to be activated with the common command `*RCL <number>`.

6.10.2 Extensions for user files

The following table lists all available file extensions for user files. The currently available files on the instrument depend on the installed options.

Table 6-1: List of the automatically assigned file extensions in the instrument

Function	List type	Contents	File suffix
Instrument State	Settings	Instrument settings	`*.savrcltxt`
"User Correction"	List	User-defined level correction values	`*.uco`
		Export Data	`*.txt` or `*.csv`
"List Mode"	List	User-defined frequency/level value pairs	`*.lsw`
		Export Data	`*.txt` or `*.csv`
"Pulse Train List"		User-defined offtime/ontime/repetition values	`*.pulstrn`
SMZ Settings	Settings	Data (firmware) of a connected SMZ frequency multiplier	`*.efmfirm`
NRP Settings	Settings	NRP Settings	`*.nrp`

6.10.3 Examples

In these examples, the current instrument setting is stored in the file `test.savrcltxt` in the directory `/var/user/.`.

Storing and Loading Current Settings

1. Store the current setting in an intermediate memory with the number 4. This setting can be called using command `*RCL` and the associated number of the memory, for example `*RCL 4`.
```
*SAV 4
```
2. To store the settings in a file in a specific directory, specify the complete path.
```
MMEM:STOR:STAT 4,"/var/user/test.savrcltxt"
```

3. To store the settings in a file in the default drive, set the default drive and specify only the file name.
   ```
   MMEM:CDIR '/var/user/'*SAV 4
   MMEM:STOR:STAT 4,"test.savrcltxt"
   ```
4. Load the file `test.savrcltxt` in the `user` directory.
   ```
   MMEM:LOAD:STAT 4,'/var/user/test.savrcltxt'
   ```
5. Activate the instrument setting of the file `test.savrcltxt`.
   ```
   *RCL 4
   ```

Working with Files and Directories

1. Read out all files in the specified directory.
   ```
   MMEM:CAT? '/usb/user'
   ```
   Response: `127145265,175325184,"test,DIR,0","temp,DIR,0", "readme.txt,ASC,1324","state.savrcltxt,STAT,5327", "waveform.wv,BIN,2342"`

   the directory `/usb/user` contains the subdirectories `test` and `temp` as well as the files `readme.txt`, `state.savrcltxt` and `waveform.wv` which have different file types.

   Tip: To query only the subdirectories of the current or specified directory, perform:
   ```
   MMEM:DCAT? '/usb/user'
   ```
   Response: `'test', 'temp'`

   To query only the number of subdirectories in the current or specified directory, perform:
   ```
   MMEM:DCAT:LENG? '/usb/user'
   ```
   Response: `2`
2. To query the number of files in the current or specified directory, perform:
   ```
   MMEM:CAT:LENG? '/usb/user'
   ```
   Response: `3`
3. Create a new subdirectory for mass memory storage in the specified directory.
   ```
   MMEM:MDIR '/usb/new'
   ```
4. Copy the file `state` to a new file.
   ```
   MMEM:COPY '/var/user/state.savrcltxt','/usb/new'
   ```
5. Rename the file `state`.
   ```
   MMEM:MOVE 'state.savrcltxt','state_new.savrcltxt'
   ```
6. Remove the `test` directory.
   ```
   MMEM:RDIR '/usb/test'
   ```