6.10.4 Remote control commands

:MMEMory:CATalog? <path>

Returns the content of a particular directory.

Query parameters:

<path>   string

String parameter to specify the directory.
If you leave out the path, the command returns the contents of
the directory selected with :MMEMory:CDIRectory.
The path may be relative or absolute.

Return values:

<UsedDiskSpace>   Byte size of all files in the directory.

<FreeDiskSpace>   Remaining disk space in bytes.

<FileInfo>   <NameFileN>,<SuffixFileN>,<SizeFileN>

List of files, separated by commas

<NameFileN>
Name of the file.

<SuffixFileN>
Type of the file. Possible suffixes are: ASCii, BINary, DIRectory

<SizeFileN>
Size of the file in bytes.

Example: See "Working with Files and Directories" on page 307.

Usage: Query only

Manual operation: See "Directory, File List and File Name" on page 129

:MMEMory:CATalog:LENGth? <Path>

Returns the number of files in the current or in the specified directory.

Query parameters:

<Path>　　　　　　　　　　string

String parameter to specify the directory. If the directory is omitted, the command queries the content of the current directory, queried with `:MMEMory:CDIRectory` command.

Return values:

<FileCount>　　　　　　　integer

Number of files.

Example:　　　　　　　　See "Working with Files and Directories" on page 307.

Usage:　　　　　　　　　Query only

:MMEMory:CDIRectory <Directory>

Changes the default directory for mass memory storage. The directory is used for all subsequent `MMEM` commands if no path is specified with them.

Parameters:

<Directory>　　　　　　　<directory_name>

String containing the path to another directory. The path can be relative or absolute.
To change to a higher directory, use two dots '..' .

Example:　　　　　　　　See "Working with Files and Directories" on page 307.

Usage:　　　　　　　　　SCPI confirmed

Manual operation:　　　See "Directory, File List and File Name" on page 129

:MMEMory:COPY <SourceFile>[,<DestinationFile>]

Copies an existing file to a new file. Instead of just a file, this command can also be used to copy a complete directory together with all its files.

Setting parameters:

<SourceFile>　　　　　　string

String containing the path and file name of the source file

<DestinationFile>　　　string

String containing the path and name of the target file. The path can be relative or absolute.
If <DestinationFile> is not specified, the <SourceFile> is copied to the current directory, queried with the `:MMEMory:CDIRectory` command.
Note: Existing files with the same name in the destination directory are overwritten without an error message.

Example:　　　　　　　　See "Working with Files and Directories" on page 307.

Usage:　　　　　　　　　Setting only
SCPI confirmed

Manual operation: See "Copy " on page 133

:MMEMory:DATA <Filename>, <BinaryBlock>
:MMEMory:DATA? <Filename>

The setting command writes the block data `<BinaryBlock>` to the file identified by `<Filename>`.

Set the GPIB-bus terminator to `EOI` to ensure correct data transfer.

The query command transfers the specified file from the instrument to the GPIB-bus and then on to the controller. It is important to ensure that the intermediate memory on the controller is large enough to take the file. The setting for the GPIB-bus terminator is irrelevant.

Tip: Use this command to read/transfer stored instrument settings or waveforms directly from/to the instrument.

Parameters:

<BinaryBlock>	#<number><length_entry><data>
	#: Hash sign; always comes first in the binary block
	<number>: the first digit indicates how many digits the subsequent length entry has
	<length_entry>: indicates the number of subsequent bytes
	<data>: binary block data for the specified length.
	For files with a size with more than nine digits (gigabytes), the instrument allows the syntax #(<Length>), where <Length> is the file size in decimal format.

Parameters for setting and query:

<Filename>	string
	String parameter to specify the name of the file.

Example:	`MMEMory:DATA '/var/user/test.txt',#15hallo`
	Writes the block data to the file `test.txt`.
	The digit `1` indicates a length entry of one digit; the digit `5` indicate a length of the binary data (`hallo`) in bytes.
	`MMEMory:DATA? '/var/user/test.txt'`
	Sends the data of the file `test.txt` from the instrument to the controller in the form of a binary block.
	`Response: #15hallo`

Usage:	SCPI confirmed

:MMEMory:DCATalog? <path>

Returns the subdirectories of a particular directory.

Query parameters:

<path>	String parameter to specify the directory. If the directory is omitted, the command queries the content of the current directory, queried with :MMEMory:CDIRectory command.

Return values:

<Catalog>	<file_entry>
	Names of the subdirectories separated by colons. The first two strings are related to the parent directory.

Example: See "Working with Files and Directories" on page 307.

Usage: Query only

:MMEMory:DCATalog:LENGth? [<Path>]

Returns the number of subdirectories in the current or specified directory.

Query parameters:

<Path>	String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory, to be queried with :MMEMory:CDIRectory command.

Return values:

<DirectoryCount>	integer
	Number of parent and subdirectories.

Example: See "Working with Files and Directories" on page 307.

Usage: Query only

:MMEMory:DELete <Filename>

Removes a file from the specified directory.

Setting parameters:

<Filename>	string
	String parameter to specify the name and directory of the file to be removed.

Example: See "Working with Files and Directories" on page 307.

Usage: Event
SCPI confirmed

Manual operation: See "Cut" on page 133

:MEMory:HFRee?

Returns the used and available memory in Kb.

Return values:

<TotalPhysMemKb>	integer
	Total physical memory.
<ApplicMemKb>	integer
	Application memory.

<HeapUsedKb>	integer
	Used heap memory.
<HeapAvailableKb>	integer
	Available heap memory.
Usage:	Query only

:MMEMory:LOAD:STATe <SavRclStateNumb>, <file_name>

Loads the specified file stored under the specified name in an internal memory.

After the file has been loaded, the instrument setting must be activated using an `*RCL` command.

Setting parameters:

<SavRclStateNumb>	Determines to the specific <number> to be used with the `*RCL` command, e.g. `*RCL 4`.
<file_name>	String parameter to specify the file name with extension `*.savrcltxt`.
Example:	See "Storing and Loading Current Settings" on page 306.
Usage:	Setting only
Manual operation:	See "Recall" on page 131

:MMEMory:MDIRectory <Directory>

Creates a subdirectory for mass memory storage in the specified directory. If no directory is specified, a subdirectory is created in the default directory. This command can also be used to create a directory tree.

Setting parameters:

<Directory>	string
	String parameter to specify the new directory.
Example:	See "Working with Files and Directories" on page 307.
Usage:	Event
Manual operation:	See "Create New Directory" on page 133

:MMEMory:MOVE <SourceFile>, <DestinationFile>

Moves an existing file to a new location or, if no path is specified, renames an existing file.

Setting parameters:

<SourceFile>	string
	String parameter to specify the name of the file to be moved.

<DestinationFile>	string
	String parameters to specify the name of the new file.
Example:	See "Working with Files and Directories" on page 307.
Usage:	Event
	SCPI confirmed
Manual operation:	See "Rename " on page 133

:MMEMory:MSIS <Msis>

Defines the drive or network resource (in the case of networks) for instruments with windows operating system, using `msis` (MSIS = Mass Storage Identification String).

Note: Instruments with Linux operating system ignore this command, since Linux does not use drive letter assignment.

Usage:	SCPI confirmed

:MMEMory:RDIRectory <Directory>

Removes an existing directory from the mass memory storage system. If no directory is specified, the subdirectory with the specified name is deleted in the default directory.

Setting parameters:

<Directory>	string
	String parameter to specify the directory to be deleted.
Example:	See "Working with Files and Directories" on page 307.
Usage:	Event

:MMEMory:STORe:STATe <savrcl_state_nr>, <file_name>

Stores the current instrument setting in the specified file.

The instrument setting must first be stored in an internal memory with the same number using the common command `*SAV`.

Setting parameters:

<savrcl_state_nr>	Corresponds to the specific `<number>` defined with the `*SAV` command, e.g. `*SAV 4`.
<file_name>	String parameter to specify the file name with extension `*.savrcltxt`.
Example:	See "Storing and Loading Current Settings" on page 306.
Usage:	Event
Manual operation:	See "Save" on page 129