

fmsx

COLLABORATORS

	<i>TITLE :</i> fmsx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 14, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	fmsx	1
1.1	"	1
1.2	"	3
1.3	"	4
1.4	"	4
1.5	"	5
1.6	"	5
1.7	"	6
1.8	"	6
1.9	"	7
1.10	"	7
1.11	"	7
1.12	"	8
1.13	"	8
1.14	"	8
1.15	"	8
1.16	"	9
1.17	"	9
1.18	"	10
1.19	"	10
1.20	"	10
1.21	"	11
1.22	"	11
1.23	"	11
1.24	"	12
1.25	"	12
1.26	"	13
1.27	"	13
1.28	"	13
1.29	"	13

1.30 "	14
1.31 "	14
1.32 "	14
1.33 "	15
1.34 "	15
1.35 "	15
1.36 "	16
1.37 "	16
1.38 "	17
1.39 "	17
1.40 "	17
1.41 "	18
1.42 "	19
1.43 "	19
1.44 "	19
1.45 "	20
1.46 "	20

Chapter 1

fmsx

1.1 "

fMSX Amiga ARexx commands

General information

fMSX and ARexx

Obsolete commands

Alphabetical list of commands

AHI_MixFreq

AHI_SoundID

Bank

Cartridgel

DriveA

DriveB

Drives

EnableExtFont

EnableExtROMs

EnableKanji

FMROMSlot

HideTitlebar

InstThreshold

InterruptPeriod

InterruptMode
InterruptRate
JapaneseID
Joystick1
LockDrives
Memory
MessageLine
Mode
MSX1ROM
MSX2ROM
MSX2SubROM
MSX2PlusROM
MSX2PlusSubROM
MSXFont
MSXSpeed
MSXVersion
MSXWindow
PlugAndPlay
PublicScreen
PublicScreenName
Quit
RefreshWhenActive
Reset
ROMType1
ROMType
SavePrefsOnQuit
SCCDisable
SCCPlus

ScreenMode
 SkippedFrames
 RefreshCycle
 SoundMode
 TapeAppend
 TapeIndex
 TemporaryDisk
 Version
 Window

1.2 "

fMSX and ARexx

Since version 1.4 fMSX has an ARexx port. This document lists all available commands and their options. It assumes that the reader already possesses some knowledge about ARexx, but don't worry if you don't: I know almost nothing about it either and I am supposed to write this stuff. Besides, documentation describing ARexx is widely available on Aminet and other places.

The ARexx portname is different for each running copy of fMSX. The first copy is called FMSX.0, the second copy FMSX.1, and so on. The example scripts assume you are running only one copy. If you wish to address multiple copies you'll have to modify the scripts yourself.

Since fMSX 2.0 the same command interpreter is used for ARexx commands, the preferences file, and the commands stored in the icons of cartridges and disks. Not all commands make sense as ARexx commands.

Generally speaking all commands that set some attribute of fMSX also return the current setting of that attribute, and can be used without arguments just to obtain that setting. As an example, take a look at the mode command. It can be used to pause fMSX like this:

```
mode pause
```

However, if you want to toggle the mode between running and pause you would have to obtain the current mode and invert it, like this:

```

options results      /* We want to hear about results */
address FMSX.0      /* Talk to fMSX */
mode                /* Find the current mode */
if result = run then
  mode pause        /* Pause if we were running */
else
  mode run          /* Run if we were pausing, or in music mode */

```

More examples are available in the ARexx directory of the fMSX distribution.

If you want to see more ARexx commands added, or if you have written your own scripts and want to share them with the world, please contact the author.

1.3 "

Obsolete commands

A number of commands that were available in fMSX 1.4 are no longer available in fMSX 2.0. This section lists which commands have been removed and why.

AnchorScreen:	no longer required, fully automatic.
AutoSavePrefs:	replaced by "SavePrefsOnQuit".
DeviceA:	replaced by "DriveA".
DeviceB:	replaced by "DriveB".
DirectColorLoad:	feature is obsolete.
DoubleBuffer:	no longer required, fully automatic.
LoadCartridge1:	replaced by "Cartridge1".

The following commands are all related to the dynamic recompiler of fMSX 2.0, and are obsolete since fMSX 2.1.

DAAEmulation
FlushZ80Cache
InlineLength
SpeedControl
TimingInfo
UpdateStats
Z80CacheSize
Z80HashTableSize

1.4 "

AHI_MixFreq

Purpose: to set the AHI mixing frequency.

Format: [<number>]

Available: 2.0

Description: this command changes the mixing frequency used by AHI. Higher mixing frequencies use more processing power, causing the entire system to run a bit slower. Lower frequencies do not sound as good as higher frequencies, though.

If you have the processing power, put it on the highest possible setting. The highest possible value differs from Amiga to Amiga, depending on the presence of a soundcard and the availability of 30KHz screenmodes.

1.5 "

AHI_SoundID

Purpose: to set the sound mode used by AHI.

Format: [<number>]

Available: 2.0

Description: this command changes the soundmode used by AHI. Choose a mono mode, if possible: fMSX only generates mono sound anyway.

Each mode has a unique number. Which modes are available depends on what type of sound hardware you have installed in your Amiga.

1.6 "

Bank

Purpose: to set the value of a single bank of battery backed memory.

Format: <number>[<number><number><number><number>]

Available: 2.0

Description: this command changes the value of a single bank of battery backed memory. The first attribute, which is required, is the bank number. This is a value between 0 and 15. The optional other four attributes are the bytes that must be stored in this bank.

The MSX stores various types of information in the battery backed memory, including the password selected by the user, his prompt of choice, the starting screenmode and width that screen, the starting colors, and the default settings of keyclick and function key visibility. Although you could conceivably set them using this command it was really not provided to be used through ARexx. Instead it is meant to allow the saving of battery backed memory in the preferences file.

To force the MSX to boot in the way you like it, first change all the attributes you want to change, then use the SETSCREEN command. As an example, try this:

```
SCREEN 0
WIDTH 80
COLOR 15,0
KEY OFF
SETSCREEN
SETPROMPT "Files"
```

After saving this as your default preferences, fMSX will continue to boot with these attributes.

Note that all of these commands, as well as the entire battery backed memory

system, require MSX2 or better to work.

1.7 "

Cartridge1

Purpose: to insert a cartridge into fMSX.

Format: [<filename>]

Available: 2.0

Description: This command 'inserts' a virtual cartridge into fMSX. It will not be activated until the MSX is reset though.

Description: this command specifies a cartridge name for use with fMSX. The cartridge will be loaded when fMSX is reset.

If a cartridge is loaded the return value is the name of that cartridge, including the path (if any). If no cartridge is loaded the return value is the path to the cartridge directory as specified in the control window.

Note that certain operations on cartridges (such as saving cartridge-specific information in the icon) use the filename you specify here. If you change this name and then save the icon you will save the icon to the wrong location.

Also note that after a reset fMSX will set several controls based on information found in the icon of the cartridge, if any, overriding any changes you may have made before.

The '1' at the end of the commandname means that the cartridge will be loaded into MSX-slot 1. In the future support may (once again) be added for using two cartridges at the same time. A second cartridge will have to be loaded into slot 2.

1.8 "

DriveA, DriveB

Purpose: to insert a new disk in an MSX drive.

Format: [<filename>|<directoryname>|<devicename>]

Available: 2.0

Description: These commands insert a new disk in MSX drive A: or B:, respectively. A 'disk' can be either many things: a real floppydisk, such as PC0:, or a diskimage, or a directory that contains MSX files.

As with a real MSX, you shouldn't insert disks while the MSX is using that drive. If you do so anyway you risk corrupting the data on the disk.

1.9 "

Drives

Purpose: to change the number of MSX drives used by fMSX.

Format: [<none>|<1>|<2>]

Available: 1.4

Description: this command changes the number of drives used by the MSX. An MSX can use up to two drives, known as A: and B:. In most cases you will want to use one drive (using a second drive uses up more MSX RAM, which will cause some programs to fail).

Note that this setting takes effect when you reset the MSX.

1.10 "

EnableExtFont

Purpose: to change external font loading.

Format: [On | Off]

Available: 1.4

Description: this command changes whether fMSX will use the internal ROM font of the currently loaded MSX ROMs or an external font of your choosing. See

MSXFont
for more information.

Note that this setting takes effect when you reset the MSX.

1.11 "

EnableExtROMs

Purpose: to change external ROM loading.

Format: [On | Off]

Available: 1.4

Description: this command changes whether fMSX will use externally loaded ROMs or the internal ones. See

MSXROM
for more information.

Note that this setting takes effect when you reset the MSX.

1.12 "

EnableKanji

Purpose: to change whether Kanji ROMs are loaded.

Format: [On | Off]

Available: 1.4

Description: this command changes whether fMSX will use the Kanji ROM. See KanjiROM for more information.

Note that this setting takes effect when you reset the MSX.

1.13 "

FMROMSlot

Purpose: sets the slot used by the FM-Pac ROM.

Format: [None | Cartridge | Disk]

Available: 2.0

Description: this command sets the slot used by the FM-Pac ROM.

1.14 "

HideTitlebar

Purpose: to hide or show the titlebar of the MSX screen.

Format: [On | Off]

Available: 1.4

Description: This command hides or reveals the titlebar of the MSX screen. When it is turned on you can easily drag the screen down. However, turning it off makes for a completer MSX experience.

1.15 "

InstThreshold (was: InterruptPeriod)

Purpose: to change the instruction threshold.

Format: [<number>]

Available: 2.0

Description: this command changes the instruction threshold. The instruction threshold is the minimum number of branches taken by the Z80 between two interrupts. On slow processors interrupts may happen so quickly that the running program does not have enough time to finish normal processing between interrupts. This may cause crashes. Setting the instruction threshold to a higher value may solve these crashes.

An example of a game that needs a high interrupt period (at least, on 68030-based machines) is Maze of Galious. When the interrupt period drops below 1500 it stops redrawing the screen, leaving large black gaps.

Note that only multiples of 100 are accepted for the instruction threshold. Other values will be rounded to the next lower multiple of 100.

Acceptable values for the instruction threshold lie between 0 and 5000.

The obsolete alternative command, InterruptPeriod, is still accepted for reasons of backwards compatibility.

1.16 "

InterruptMode

Purpose: to set the current interrupt mode.

Format: [PAL | NTSC | Custom]

Available: 2.0

Description: sets the interrupt mode to PAL (50 interrupts per second), NTSC (60 interrupts per second), or custom (10 to 100 interrupts per second).

1.17 "

InterruptRate

Purpose: to set the custom interrupt rate.

Format: [<number>]

Available: 2.0

Description: this command sets the interrupt rate when the interrupt mode is "custom". The interrupt rate determines almost directly how fast a game runs, and in many games the interrupt rate is enough to accurately control the speed.

1.18 "

JapaneseID

Purpose: to change the country ID in the ROMs.

Format: [On | Off]

Available: 1.4

Description: this command changes the way the country ID bytes in the MSX ROMs are patched. This can be useful because some programs work differently when they find Japanese ID codes (for example: Penguin Adventure and the Nemesis series).

Note that this setting takes effect when you reset the MSX.

1.19 "

Joystick1

Purpose: to set joystick 1 to mouse or joystick.

Format: [<mouse> | <joystick>]

Available: 2.0

Description: this command sets MSX joystick 1 to be either the normal Amiga mouse, or the Amiga joystick. A few MSX programs support the use of a mouse; note that in almost all cases they check for the mouse only during startup, so you must make sure the mouse is selected before the program becomes active.

1.20 "

LockDrives

Purpose: to lock or unlock the drives used by fMSX.

Format: [On | Off]

Available: 1.4

Description: this command locks or unlocks the drives used by fMSX. When locked the drives can be used by the emulated MSX, but not by the Amiga. When unlocked the situation is reversed: the MSX cannot access the drives but the Amiga can.

From the MSX point of view unlocking is equivalent to removing the disk from the drive. As might be expected, you should not do this while the disk is being accessed.

1.21 "

Memory

Purpose: to change the amount of memory used by the MSX.

Format: [64KB | 128KB | 256KB | 512KB | 1MB | 2MB | 4MB]

Available: 1.4

Description: this command changes the amount of memory used by the MSX. An MSX can use up to 4MB per slot, and fMSX supports RAM in one slot.

Most programs will feel perfectly fine when using 256KB of RAM. Only a handful need 512KB, and I cannot name any that need more than that (apart from RAM disks and the like).

Note that this setting takes effect when you reset the MSX.

1.22 "

MessageLine

Purpose: to set a user message.

Format: <message>

Available: 2.0

Description: this command sets a single-line user message. The message will appear in the main window.

1.23 "

Mode

Purpose: to run or pause fMSX, or put it in music mode.

Format: [RUN | PAUSE | MUSIC]

Available: 1.4

Description: this command sets the mode of fMSX. fMSX has three modes of operation:

Mode RUN

fMSX is executing an MSX program. It will take 100% CPU time, split between the MSX emulation and redrawing the screen. Because the MSX emulation is running at priority -1 it has no impact on other programs running on the Amiga.

Mode PAUSE

fMSX is suspended, and takes no CPU time whatsoever.

Mode MUSIC

fMSX is running but does not redraw the screen. This frees up more CPU cycles for the actual MSX emulation which runs faster as a result. Although you cannot see anything you can hear whatever sounds it may generate, which is great for running MSX music demo's.

1.24 "

MSXFont

Purpose: to change the path to the externally loaded MSX font.

Format: [<filename>]

Available: 1.4

Description: this command tells fMSX what external MSX font you want to use, if any. This can be used to change to a Japanese or Cyrillic font (which can be useful), or simply to a 'cool' font such as the italic font.

Externally loaded MSX fonts are not compatible with Amiga bitmap fonts. If you try to load an external Amiga bitmap font your MSX screen will be unreadable.

Several fonts can be found in the MSXFonts directory in the fMSX distribution.

Note that this setting takes effect when you reset the MSX.

1.25 "

MSX1ROM, MSX2ROM, MSX2SubROM, MSX2PlusROM, MSX2PlusSubROM

Purpose: to change the path to the one of the MSX ROMs.

Format: [<filename>]

Available: 1.4

Description: this command sets the path to one of the MSX ROMs. You may want to use your own ROMs with fMSX instead of the ones that are built in. If you want to do that you will need to specify the correct paths.

The MSX1 system only needs one ROM, which contains both BIOS and BASIC. However, the MSX2 system needs a second ROM called the sub ROM, which contains the extra BASIC commando's supported by the MSX2. The BIOS/BASIC ROM is 32KB for both MSX1 and MSX2, and the sub ROM is 16KB.

Note that this setting takes effect when you reset the MSX.

1.26 "

MSXSpeed

Purpose: to change the current MSX speed.

Format: [<number>]

Available: 2.0

Description: this command changes the relative speed of the emulated MSX compared to a real MSX when running in Enhanced Speed Control mode.

1.27 "

MSXVersion

Purpose: to change the MSX ROMs used by fMSX.

Format: [MSX1 | MSX2 | MSX2+]

Available: 1.4

Description: this command changes the ROMs used by fMSX. Although all MSX1 programs work on MSX2 as well you may prefer to use MSX1 ROMs when possible because the MSX1 bootsequence is much faster than the one for MSX2.

This command only affects the ROMs used by fMSX. Hardware-wise fMSX always emulates an MSX2 machine.

Note that this setting takes effect when you reset the MSX.

1.28 "

MSXWindow

Purpose: specifies the coordinates of the MSX window.

Format: [<number> <number>]

Available: 2.0

Description: this command species the coordinates of the MSX window (that's the window used when running on a public screen).

This command is meant for use in the preference file, rather than ARexx.

1.29 "

PlugAndPlay

Purpose: to change the way fMSX reacts to cartridge insertions.

Format: [On | Off]

Available: 1.4

Description: this command changes the way fMSX reacts to the insertion of cartridges. When plug'n'play is turned on fMSX will reboot after a cartridge is inserted (when it is turned off nothing will happen).

Note that only fMSX only reboots after interactive cartridge insertions (ie. cartridges insert through GUI-commands). If you want to reboot after inserting a cartridge through ARexx you will have to call

Reset

.

1.30 "

PublicScreen

Purpose: puts fMSX on a public or custom screen.

Format: [On | Off]

Available: 2.0

Description: this command turns running on public screens on or off.

fMSX can only run on public CyberGfx screens.

1.31 "

PublicScreenName

Purpose: specify the name of the public screen to run on.

Format: [<screenname>]

Available: 2.0

Description: this command specifies the name of the public screen to run on. Only CyberGfx screens are possible.

1.32 "

Quit

Purpose: to quit fMSX.

Format: -

Available: 1.4

Description: this command quits fMSX.

1.33 "

RefreshWhenActive

Purpose: to change screenrefreshing when it is inactive.

Format: [On | Off]

Available: 1.4

Description: With this command you can instruct fMSX not to refresh the MSX screen when it is inactive. This frees up processor power for other applications or the main emulation while you are working in another window.

1.34 "

Reset

Purpose: to reset the emulated MSX.

Format: -

Available: 1.4

Description: this command resets the MSX machine emulated by fMSX. You need to reset when you want to leave the current game, or if you want to (de-)activate certain features.

1.35 "

ROMType1, ROMType

Purpose: to set the ROM type for the currently loaded cartridge.

Format: [SCC1 | SCC2 | KONAMI1 | KONAMI2 | ASCII1 | ASCII2]

Available: 1.4

Description: this command sets the ROM type for the currently loaded cartridge. Setting the ROM type only affects megaROMs (cartridges that are bigger than 32KB). 16KB and 32KB ROMs do not have a ROM type.

It is necessary to set the correct ROM type for a megaROM to function. If you do not know the ROM type for a particular megaROM you have no choice but to try them all. However, a list of ROM types is available here [games.guide/main](#)).

The '1' at the end of the commandname means that the rom type will be set for the cartridge in MSX-slot 1. In the future support may (once again) be added for using two cartridges at the same time. A command called ROMType2 will then be added as well.

For reasons of backwards compatibility the command "ROMType" is also accepted.

1.36 "

SavePrefsOnQuit

Purpose: to change the automatic saving of preferences.

Format: [On | Off]

Available: 2.0

Description: this command changes the automatic saving of preferences.

Apart from fMSX settings the preferences file also contains the values stored in the MSX battery-backed RAM. These include the initial screenmode, color settings, whether function keys are on or off, prompt, password, etc. A list of configuration options can be found here.

On a real MSX the battery backed RAM also contains time and date values, but in fMSX these are always read from the Amiga's clock. fMSX never tries to change the Amiga clock - too many MSX programs reset the clock for some reason.

1.37 "

SCCDisable

Purpose: to change SCC channel disabling.

Format: [On | Off]

Available: 1.4

Description: with this command you can select whether fMSX ignores or respects the SCC disable settings. In most games, ignoring those settings gives the best soundquality, but in some games (such as Parodius) the soundquality is much better when this feature is turned on.

This command only affects the SCC and SCC/PSG soundmodes.

1.38 "

SCCPlus

Purpose: to change SCC+ emulation.

Format: [On | Off]

Available: 1.4

Description: this command enables or disables SCC+ emulation. SCC+ is a soundchip used in some Konami games, and is an improved version of the SCC soundchip. When SCC+ emulation is enabled, (normal) SCC emulation is disabled and vice versa.

Enabling SCC+ only makes sense when you are using the SCC or PSG/SCC soundmodes.

As far as I know, SCC+ is only used in The Snatcher and S.D. Snatcher.

1.39 "

ScreenMode

Purpose: to set a screenmode.

Format: <number> [<number>]

Available: 2.0

Description: this command changes the display ID of one of the supported screenmodes. The first parameter is the screenmode to be changed, the second parameter is the display ID for this screenmode.

1.40 "

SkippedFrames (was: RefreshCycle)

Purpose: to change the number of frames skipped.

Format: [<number>]

Available: 2.0

Description: this command sets the number of frames skipped by fMSX. This directly influences the speed of the emulation: the more often the screen is redrawn, the slower the emulation runs. On an 68060 it is possible to run without skipping any frames, other machines may have to skip one or more to keep speed up.

The obsolete version of this command, RefreshCycle, is still accepted for reasons of backwards compatibility.

1.41 "

SoundMode

Purpose: to change the way fMSX emulates sound.

Format: [Off | PSG | SCC | PSGSCC | PSGFMPac | FMPac]

Available: 1.4

Description: this command changes the way fMSX emulates sound. Currently five methods are provided:

Off

fMSX is not playing any sound, either because that is what the user wanted or because it could not allocate the sound hardware.

PSG

fMSX is emulating the native MSX soundchip, called the Programmable Sound Generator or PSG. Although it allocates the sound hardware before use it hits the hardware for producing sound. This mode is in many ways the best fMSX has to offer: it is fast, sounds exactly like a real MSX, and has support for digitized samples (through volume modulation of the noise channel).

SCC

fMSX is emulating a soundchip found in some Konami games. It produces much better sound than the PSG, but offers some features that cannot easily be emulated on the Amiga (such as five-channel sound).

Examples of games that support SCC sound are:

MSX1 games	MSX2 games
Salamander	Space Manbow
Nemesis 2	Metal Gear 2
Nemesis 3	Quarth
King's Valley 2	Gryzor
F1 Spirit	King's Valley 2 MSX2
Parodius	The Snatcher
	S.D. Snatcher

PSGSCC

fMSX is emulating both the PSG and SCC soundchips. This is done with the help of AHI, a system for playing up to 128 channels of sound using nothing more than the standard Amiga audio hardware and a fast processor.

In this mode fMSX runs a lot slower because AHI is mixing samples in the background. The soundquality is not quite as good as with the normal PSG or SCC modes, but at least you can listen to the full musical scores which tend to be rather spectacular for SCC games.

PSGFMPac

In this mode fMSX is attempting to emulate the FM-Pac. Currently this is a very experimental mode, which sounds crap most of the time. I hope to improve this in the future. This mode requires AHI as well.

FMPac

In this mode fMSX uses the Concierto soundcard (which is an add-on to the Picasso IV graphics card) to emulate the FM-PAC. The emulation is spot-on; not surprising, since the OPL3 used in the Concierto is a descendant of the OPLL used in the FM-PAC.

A Concierto soundcard is required to use this mode.

1.42 "

TapeAppend

Purpose: to turn tape appending on or off.

Format: [On | Off]

Available: 2.0

Description: this command turns auto-appending on tape on or off.

1.43 "

TapeIndex

Purpose: specifies the tape index to be used.

Format: [<filename>]

Available: 2.0

Description: this command specifies the tape index to be used.

1.44 "

TemporaryDisk

Purpose: to set the location of the temporary disk.

Format: [<filename>]

Available: 2.0

Description: this command sets the location of the temporary disk. The temporary disk is used when converting a directory of files into a disk image on the fly. Be sure to put it in a place where you can store 720KB

without problems.

1.45 "

Version

Purpose: to obtain the version number of fmsx.

Format: [MINOR | MAJOR]

Available: 1.4

Description: this command returns the version number of fmsx. You can use the version number to decide whether fmsx will support certain operations.

When used with the MINOR (or MAJOR) argument, only the minor (or major) version number is returned.

When used without arguments, both major and minor version numbers will be returned, seperated by a dot.

1.46 "

Window

Purpose: to set the shape and position of a window.

Format: <number> [<number> <number> <number> <number>]

Available: 2.0

Description: this command changes the position and size of a window. The first parameter is the number of the window to be changed, and the other four are the top, left, width, and height of that window.

Window numbers are:

- 1 main window
- 2 system settings window
- 3 video settings window
- 4 CPU settings window
- 5 tape window
- 6 paths window
- 7 sound settings window
- 8 cheat window

Note that this command is meant for use in the preference file.
