MainActor's Rexx Commands

MainActor features advanced scripting capabilities. You can automate and custom program nearly every aspect of MainActor.

We did choose **Rexx** for the scripting language as we think it is the best programming language for scripting, it is simple but very powerful.

This document lists and explains all Rexx commands which are MainActor specific, i.e. you can use to access the MainActor functionality from inside Rexx scripts.

Rexx is built into the OS/2 operating system, under Windows we have licensed the **Rexx** processor from **Enterprise Alternatives**. If you you are interested to use Rexx (under Windows) outside of MainActor, please have a look at their WebSite: http://www.winrexx.com**.**

The MainActor specific commands, called from a Rexx script, look like this:

```
<RESULT>=Command( <ARG1>, <ARG2>, ... )
```

We have devided the overview of the commands into several sections. The **Action Commands** section lists all commands which perform a certain action inside MainActor, the **Information Commands** section lists all commands which request specific information from MainActor (like the size of a frame) and finally the **Configuration Commands** section lists the commands for changing various global configuration settings from MainActor.

Action Commands

This chapter lists the commands which perform a certain action inside MainActor. Action commands return 0 if the operation was completed successfully or an error code > 0 if not. For example, most of the commands require that a project was already loaded into MainActor and will return an error code otherwise.

CloseProject()

Closes the currently selected project.

DeselectAll()

Deselects all frames in the frame list.

DeselectFrame(<FRAME>)

Deselects the specified frame in the frame list.

DisableFrameContainer()

Disables the frame container. Very useful if a larger number of commands are issued which require updates to the frame container (especially the **MovePicture** and **SwapPictures** commands). This command disables the necessary updates to the user interface and therefore speeds up operation. Don't forget to call **EnableFrameContainer** when you are ready with the frame container operations.

EnableFrameContainer()

Enables updates to the frame container, after it was disabled with **DisableFrameContainer**.

GotoFrame(<FRAME>)

Jumps to a certain frame in the frame list, i.e. ensures that the frame is visible. The frame will not be selected or otherwise altered, however it becomes the cursored frame which will be for example shown with the **ShowFrame** command.

LoadPictureList(<BASEPICTURE>, <PICTURES>)

Loads a picture list. <BASEPICTURE> specifies the first picture in the list to load, <PICTURES> specifies the number of picture to load from the base picture. If <PICTURES> is 0, all pictures in the list will be loaded.

Example: LoadPictureList("c:.050", 3) will load in "c:.050", "c:.051" and "c:.052".

LoadProject(<NAME>)

Loads the specified project, if **<NAME>** is an empty string (i.e. "") you will be prompted with a file-requester. LoadProject does not support DOS patterns, use **LoadProjectPattern** for this.

LoadProjectPattern(<PATTERN>)

This command accepts a valid DOS pattern and in each call loads the next available file represented in the pattern. When no more files are available an error code is returned. This command enables you to load whole directories of animations and convert them in one step.

Example: DO WHILE LoadProjectPattern("c:.*") =0 BEGIN ... END would load each file in "c:, you could place a **Save** and **CloseProject** command inside the BEGIN / END block to automatically save and close each project.

MovePicture(<FROM>, <TO>)

Moves a picture in a picture list from a position inside the list to another location inside the list.

Example: MovePicture(1, 5) would move the first frame of the list to position 5.

PlayProject()

Plays the current project.

Quit()

Closes all projects and exits MainActor.

RemovePictures()

Removes all selected pictures in the frame list. Only works for picture lists.

Save(<NAME>, <MODULE>, <CODEC>, <FRAMES>, <WIDTH>, <HEIGHT>)

Saves the current project. Here are the descriptions of the available options:

- <**NAME>**: Set the name of the new project. Use "" if you want to get a file requester.
- **<MODULE>**: Specify the module which you want to use, like "AVI" or "MPEG-I".
- **<CODEC>**: Specify the codec you want to use, you can use substrings to identify the codec, i.e. you don't have to use "Millions of Colors: Cinepak(TM) Radius" but can simply use "Cinepak".
- **<FRAMES>**: Specify if you want to save all, or only the selected frames of the project. Specify **"AllFrames"** to save all, or **"SelectedFrames"** to save the selected frames of the project.
- **<WIDTH>**: Enter the width of the project, simply use "0" to use the width of the source project.
- **<HEIGHT>**: Enter the height of the project, simply use "0" to use the height of the source project.

Example: Save("c:", "AVI", "Cinepak", "AllFrames", 0, 0) would save all frames of the current project to "c:.avi" in Cinepak format without changing the width and height of the images.

SelectAll()

Selects all frames in the frame list.

SelectFrame(<FRAME>)

Selects the specified frame in the frame list.

SelectProject(<ARG1>, <ARG2>)

Selects one of the currently loaded projects. You have two ways to select a project:

- **<ARG1>="NAME"**: If <ARG1> is "NAME", <ARG2> has to be the name of the project as displayed in the project list (like "nice.avi" or "IFF").
- **<ARG1>="NUMBER"**: If <ARG1> is "NUMBER", <ARG2> has to be the number of the project, "1" is the topmost project in the list.

Example: SelectProject("NAME", "nice.avi") would select the project with the name "nice.avi". **SelectProject("NUMBER", GetGlobalInfo("LOADEDPROJECTS")** would select the last project in the list as the call to GetGlobalInfo() returns the number of currently loaded projects.

SetLocalTimecode(<FRAMENUMBER>, < TIMECODE>)

Sets the local timecode (if suppored) of the frame specified by its frame number. <TIMECODE> has to be passed in milliseconds.

SetLocalTimecodes(<TIMECODE>)

Sets the local timecodes (if supported) of all selected frames of the project. <TIMECODE> has to be passed in milliseconds.

SetGlobalTimecode(<TIMECODE>)

Sets the global timecode (if supported) of the project. <TIMECODE> has to be passed in milliseconds.

SetInfoText(<TEXT>)

Displays the passed text in the information area of MainActor, not in the Script I/O window where the output of the "say" command of Rexx is displayed. This function should be used to set a descriptive text for operations which take a while (saving, sorting etc.).

Example: SetInfoText("Saving animation as MPEG-I")

ShowFrame()

Shows the cursored frame of the currently selected project.

Example: GotoFrame(15); ShowFrame(); would display frame 15 of the current project.

SwapPictures(<PIC1>, <PIC2>)

Swaps the positions of two pictures inside a picture list. Very useful for sorting.

Example: SwapPictures("1", GetProjectInfo("FRAMES")) would swap the first picture with the last picture, as returned by the call to GetProjectInfo.

Information Commands

This chapter lists all commands which request information from MainActor. The commands return the requested information or NULL if the information is unavailable, for example when you request audio information from a project which has no audio data.

GetFrameInfo(<FRAME>, <INFO>)

Returns information about a specific frame in the frame list of the current project. <FRAME> specifies the frame number and <INFO> can be one of the following:

- **"AUDIOSIZE"**: Returns the size of the audio data associated with the frame. If no audio data is available, NULL is returned.
- **"COMPRESSION"**: Returns the compression method of the frame.
- "**KEYFRAME**": Returns "Yes" if the frame is a key frame, "No" if not.
- **"OFFSET"**: Returns the offset of the image data of the frame inside the animation. If the frame is a picture, NULL is returned.
- **"PICTURENAME"**: Returns the full name of the picture for the frame, this is of course only possible if the current project is a picture list, otherwise NULL is returned.
- **"SIZE"**: Returns the size of the image data of the frame.
- **"TIMECODE"**: Returns the timecode of the frame in milliseconds.

Example: say GetFrameInfo("3", "SIZE") would display the size of the third frame of the current project.

GetProjectInfo(<INFO>)

Returns specific information about the current project. <INFO> can be one of the following:

- **"AUDIOMODE"**: Returns "Mono" if the project has only one audio channel and "Stereo" if it has two audio channels. Returns NULL if the project has no audio data associated.
- **"BITSPERSAMPLE"**: Returns the bitsize of every audio sample (8 or 16), if no audio data is assiociated to the project, NULL is returned.
- **"FRAMES"**: Returns the number of frames of the project.
- **"FORMAT"**: Returns the format of the project, like "AVI" or "QuickTime".
- **"FULLNAME"**: Returns the full name of the project (i.e. the name with full path information). This option returns NULL if the project is a picture list.
- **"HEIGHT"**: Returns the height of the project in pixels.
- **"NAME"**: Returns the name of the project. This is the same name which is displayed in the project list.
- **"SAMPLESPERSECOND"**: Returns the samples per second value of the project, if the project has no audio data associated, NULL is returned.
- **"SIZE"**: Returns the size of the project.
- **"TIMECODEMODE"**: Returns the type of timecode the project supports. Returns "Global Timecode" if the project supports a global timecode or "Local Timecodes" if the project supports local timecodes.
- **"TYPE"**: Returns the type of the project. "Animation" if the project is an animation, "Picture List" if the project is a picture list or " Sound" if the project is a sound.
- **"WIDTH"**: Returns the width of the project in pixels.

Example: say GetProjectInfo("FRAMES") would display the number of frames of the current project.

GetGlobalInfo(<INFO>)

Returns information about global configuration settings or information which is not directly associated with a specific project. Currently <INFO> can only be:

- **"LOADEDPROJECTS"**: Returns the number of loaded projects, "0" if no project is currently loaded.

Configuration Commands

This chapter lists the commands which set global configuration settings inside MainActor. These commands always return the old value of the configuration setting you did change, which allows you to change the setting back to its previous state at the end of your script.

SetGlobalConfig(<ARG1>, <ARG2>)

Sets global configuration settings of MainActor. <ARG1> specifies the settings name and <ARG2> the new value of the setting. <ARG1> can be one of the following:

- **<ARG1>="LANGUAGE"**: Specifies the language MainActor uses, set <ARG2> to "German" or "English". The new language will be used after the next restart of MainActor.
- **<ARG1>="MINIICONS"**: Set <ARG2> to "On" to enable mini icons for the frame container, set it to "Off" to disable the mini icons option.

SetPlaybackConfig(<ARG1>, <ARG2>)

Sets various playback settings used by display operations inside MainActor. <ARG1> specifies the settings name and <ARG2> the new value of the setting. <ARG1> can be one of the following:

- **<ARG1>="AUTOSTART"**: <ARG2> can be "On" to enable the direct starting of projects in the playback window or "Off" to disable it.
- **<ARG1>="FULLSCREEN"**: Set <ARG2> to "On" to enable fullscreen playback and to "Off" to disable it. This command is only available under Windows95/NT4.0.
- **<ARG1>="REPEATS"**: Specifies how often a project without audio shall be repeated in the playback window. <ARG2> can be number between 1 and 10000.
- **<ARG1>="SCALETOSCREEN"**: "On" will enable scaling and "Off" will disable it. Only valid in fullscreen mode. This command is only available under Windows95/NT4.0.
- **<ARG1>="TITLEBAR":** Specifies what the titlebar of the playback window shall contain, set <ARG2> to "NAME" to display the project name (which is the default), "FRAMECOUNTER" will enable a frame counter in the titlebar and "FRAMESPERSECOND" will show the frames per second rate of the project.
- **<ARG1>="USEBESTMODE"**: If <ARG2> is "On", the best screenmode will be choosen by MainActor for fullscreen playback, "Off" will result in MainActor always using the desktop mode. Only valid in fullscreen mode. This command is only available under Windows95/NT4.0.
- **<ARG1>="VOLUME"**: Sets the audio level used by MainActor, <ARG2> can be a number between 1 and 100.