

Rockit Skin Command Reference

TABLE OF CONTENTS

Overview	2
Coordinate legend:.....	2
Inside a Skin Mapper	2
Parts of the Skin Mapper	3
Drop and Move Rectangles.....	3
Skin Controls.....	4
button	4
textfield.....	4
status.....	5
indlight.....	5
edit	5
treectrl	6
listctrl	6
artwin.....	7
progress	7
meter	7
slider.....	8
knob	8
viewpeak	9
Button Command Reference Table.....	10
Text Field Reference Table.....	13
Status Field Reference Table.....	13
Indicator Light Reference Table.....	14

Overview

Rockit's display (and much functionality of the system itself), is controlled via the skin mappers, and other assorted mappers, making it almost totally user customizable. These mappers are industry standard XML based files, which are in text format, and can be easily edited with any text editor (although an XML capable editor, will make it much easier).

A skin, consists of the XML mapper, along with a folder containing a number of bitmap images, to be used for the various parts of the skin; background, buttons, etc. The first part of a line is the descriptor, for example **<background**. There are then different attributes, such as **image="DJ-100.bmp"**. Every attribute has an equal sign, followed by double quotes with the value inside the quotes. If you leave out a double quote mark – all sorts of strange things can happen. Each descriptor has to end with **/>** if on the same line, or **</background>**, if it was a multiline descriptor.

Rockit uses the same basic coordinates for all controls. Some will use width and height, some use right and bottom, depending on the type of control.

Coordinate legend:

x = left
y = top
r = right
b = bottom
w = width
h = height

In a skin mapper file, comments are designated using standard XML encoding, and look like this:

```
<!-- This is a comment. -->
```

Note that a comment must be terminated at the end.

Inside a Skin Mapper

In V4.2, the default skin is the DJ-100. In the **C:\SoftJock\RockitProDJ42\Skins** folder, you will find a file called DJ-100.xml – this is the mapper file. Under the Skins folder, you will find a folder called DJ-100 (the folder name must match the mapper file name exactly). In that folder, you will find a number of bitmaps.

If you open the DJ-100.XML file in a text editor like Notepad, the text will be very messy looking, because Notepad is not designed to understand XML, but it can still be used. If you open the file in an XML enabled editor (we use UltraEdit), it will be formatted properly, colored to highlight various sections, etc.

The first line of the mapper is the header, and is used by Rockit, as well as most text editors, to tell that this is a standard XML file, and what character set encoding it uses – do not change this line, and do not omit it.

```
<?xml version="1.0" encoding="utf-8"?>
```

The next line, is the start of the skin descriptor itself:

```
<RockitVJSkin name="DJ-100" w="1024" h="768">
```

Rockit's skinning system reads the **name="DJ-100"** attribute, to know what folder to look in for the bitmaps for the skin. At the very bottom of the mapper file, you will find the terminating tag, which will always look like this:

```
</RockitVJSkin>
```

Everything in between the two tags above, is what Rockit will interpret as part of the skin. We will now explain what each part does, and how to modify them.

Parts of the Skin Mapper

The first important part of a mapper, is the background image Rockit will display.

```
<background image="DJ-100.bmp" x="0" y="0" width="1024" height="768" type="DJ Std"/>
```

background image="DJ-100.bmp", is the main background image that all controls will reside upon.

x="0", is the starting point to display the image on the left.

y="0", is the starting point to display the image on top.

width="1024", is how wide the bitmap image is.

height="768", is the height of the bitmap image.

type="DJ Std", is optional, and will show up in the description column of the skin browser.

Note: Rockit can exist and run without a background image, although that line is still required. If you put in a non-existent filename, such as dummy.bmp, Rockit will simply fill the entire coordinate area in black. So, if you wanted a pure black background, simply put in a dummy filename. Also, the coordinates can be larger than the actual bitmap image, so if you wanted to use a smaller image on a large screen resolution, you could make the coordinates the same as your screen resolution, and anything outside the area of your image, will be black.

The next line, is the image that Rockit will display in the Skin Browser preview window, when you highlight a skin in the list.

```
<thumb image="DJ-100Thumb.bmp"/>
```

This image should be a scaled down version of your main background image. We usually take a final screenshot of the new skin running, scale it to about 300 pixels wide, and make that the thumb image.

Drop and Move Rectangles

The next two lines you will usually see, are the deck drop zones:

```
<deckAdrop x="015" y="032" r="356" b="205"/>
```

```
<deckBdrop x="669" y="032" r="1010" b="205"/>
```

Optional:

```
<deckCdrop x="000" y="000" r="000" b="000"/>
```

```
<deckDdrop x="000" y="000" r="000" b="000"/>
```

These tell Rockit that when you drag a song from a list, etc., that if they are dropped in rectangle represented by these coordinates, that Rockit should load the song directly into the respective deck. These can be zero, if no drop zones are desired/required. Be careful to avoid having these coordinates overlap a list, such as a deck queue, otherwise you may get unexpected results.

```
<windowmove x="084" y="001" r="957" b="025"/>
```

This is used to create a rectangle you can use to drag Rockit's main background window around your screen. These can be zero, if not needed. We generally make this a rectangle across the top, to simulate how standard windows programs are moved around the desktop.

Skin Controls

Skin controls in Rokit's skin mapper files, are a way for the user to do what a programmer normally does, only we take the hard work out of it, by putting them in text format, so you can easily modify. Then Rokit does all the hard work behind the scenes, as it builds the skin on startup, and translates these text controls, into highly specialized Windows controls.

The following are the individual controls that you can have on a skin, and the attributes that are allowed for each type of control. They are in no particular order of significance.

button

A button is used to place a button on the skin, that corresponds to a command in Rokit's command table. By definition, any button in Windows is a rectangle, although by making use of a transparent color, you can make the button appear to be any shape you like. See the button command reference, for available commands to map to.

Attribute	Required	Description/Comments
action	Yes	See the command reference for available values.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
image	No	The bitmap to display.
imageon	No	Bitmap to display when mouse is over the button.
hl	Yes	Highlight when mouse over. 1 = yes, 0 = no. Range 0 – 7.
tcolor	Yes	Transparent color in hex RGB format.
tooltip	Yes	Tooltip to show on mouse hover over.
font	No	Font to use if text button. Stick with standard that come with Windows for best results.
fsize	No	Font size in points.
txcolor	No	Text color in hex RGB format.
value	No	The text to draw on the button.

textfield

A textfield, is a field used to display some sort of variable information, such as the artist or title of a song in one of the decks, the current time, etc. See the text field reference table, for available fields.

Attribute	Required	Description/Comments
action	Yes	See the command reference for available values.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
bkcolor	Yes	Background color in hex RGB format.
txcolor	Yes	Text color in hex RGB format.
font	Yes	Font to use if text button. Stick with standard that come with Windows for best results.
fsize	Yes	Font size in points.
trans	Yes	Transparent background 1 = Yes, 0 = No. Overrides background color.
align	Yes	Text alignment in control: 0 = left, 1 = center, 2 = right.
value	No	The text to draw initially.

status

A status field, is used to display the status of a list, tree, etc. They are almost identical to a text field, except that Rockit handles what to display in them internally, depending on what the list contains. See the status field reference table, for available status fields.

Attribute	Required	Description/Comments
action	Yes	See the command reference for available values.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
bkcolor	Yes	Background color in hex RGB format.
txcolor	Yes	Text color in hex RGB format.
font	Yes	Font to use if text button. Stick with standard that come with Windows for best results.
fsize	Yes	Font size in points.
trans	Yes	Transparent background 1 = Yes, 0 = No. Overrides background color.
align	Yes	Text alignment in control: 0 = left, 1 = center, 2 = right.
value	No	The text to draw initially.

indlight

An indicator light, is basically like a little bulb, that turns on or off, depending on whether the corresponding feature is on or off in Rockit. It uses two images to represent the on and off state, and like buttons, can be made into any shape using a transparent color. See the indicator light reference table, for available fields.

Attribute	Required	Description/Comments
action	Yes	See the command reference for available values.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
imageon	Yes	The bitmap to display when off.
imageoff	Yes	The bitmap to display when on.
tcolor	Yes	Transparent color in hex RGB format.

edit

An edit control is used for the user to enter text into. This can be used for a search edit box. As of V4.2, there is only one available, and that is the "editsearch".

Attribute	Required	Description/Comments
action	Yes	editsearch, is currently the only available edit action.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
bkcolor	Yes	Currently not implemented.
txcolor	Yes	Currently not implemented.
font	Yes	Font to use if text button. Stick with standard that come with Windows for best results.
fsize	Yes	Font size in points.
border	Yes	Draws a single line border around box – 0 = No, 1 = Yes.
value	No	The text to draw initially.

treectrl

Tree controls are used for navigation in Rockit, and currently, there are two available – one for library navigation, and one that mimics Windows Explorer My Computer navigation.

Attribute	Required	Description/Comments
action	Yes	mylibrary or mycomputer.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
bkcolor	Yes	Background color in hex RGB format.
txcolor	Yes	Text color in hex RGB format.
font	Yes	Font to use. Stick with standard that come with Windows for best results.
fsize	Yes	Font size in points.
border	Yes	Draws a single line border around box – 0 = No, 1 = Yes.
IJ	No	Only value currently allowed is 4 – will set Vista type style with arrows.

listctrl

List controls are used for displaying songs/files. There are quite a few options available, as this is one of the most important parts of the user's interaction with Rockit. **Note:** Some of these are specialized for specific types of skins we have built, and we have not tested every possible combination of all values – experiment if you like.

Attribute	Required	Description/Comments
action	Yes	See the list below this table.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
bkcolor	Yes	Background color in hex RGB format.
txcolor	Yes	Text color in hex RGB format.
font	Yes	Font to use. Stick with standard that come with Windows for best results.
fsize	Yes	Font size in points.
border	Yes	Draws a single line border around box – 0 = No, 1 = Yes.
Header	Yes	Displays the column header (Title, Artist, etc.). 0 = No, 1 = Yes.
FH	No	Use flat header (allows more display options).
FHBkColor	No	Flat header background color in hex RGB format.
FHTxColor	No	Flat header text color in hex RGB format.
NHS	Yes	Hide Horizontal scroll bar – 0 = No, 1 = Yes.
NVS	Yes	Hide Vertical scroll bar – 0 = No, 1 = Yes.
LHI	Yes	Hide small images in list items - 0 = No, 1 = Yes.
FW	No	Full width – makes the first column the full width of the list (useful for queues).
STI	No	Show top item – always highlights the top item in the list.
HB	No	Has buddy – useful with above, and will show album art display in buddy art window.
value	No	Text to show in list if empty.
SSA	No	Show current selection always - 0 = No, 1 = Yes.
ICON	No	Will display icon view, instead of list view.
IJ	No	Varied shading options for selected item(s) – values allowed are currently 1 – 4.

Available lists:

tracklist, searchlist, historylist, requestlist, a-queue, b-queue, mx-queue.

artwin

Album cover art control. This will display the cover art if embedded within the MP3 tag of the song. If there is no cover art in the tag, Rockit will attempt to find an image on Amazon.Com to display when the song is loaded into a deck. That option can be turned on/off in the Config->General settings page.

Attribute	Required	Description/Comments
action	Yes	arta, artb, artc, artd.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, r, b	Yes	Defined in coordinate legend.
Image	Yes	Image to display if no art found in tag or on internet.
Value	Yes	Temporary file name used to display cover art.

progress

A progress control, is a visual indicator of the songs timeline. They can be a solid color, gradient color, or one of a number of predefined values we provide.

Attribute	Required	Description/Comments
action	Yes	x-progress (where x = the deck letter a – d).
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
bkcolor	Yes	Background color in hex RGB format.
pcolor1	Yes	First gradient color in hex RGB format.
pcolor2	Yes	Second gradient color in hex RGB format.
pcolor3	Yes	Third gradient color in hex RGB format.
border	Yes	Single line border - 1 = Yes, 0 = No.
value	No	Predefined (overrides colors) – H1 – H7.

meter

A meter control, is used for the input level indicators for a given deck. Very similar to a progress control above, but with the added ability of using bitmaps in place of colors or predefined values.

Attribute	Required	Description/Comments
action	Yes	x-leftmeter, x-rightmeter (where x = the deck letter a – d), cpuusage.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
bkcolor	Yes	Background color in hex RGB format.
pcolor1	Yes	First gradient color in hex RGB format.
pcolor2	Yes	Second gradient color in hex RGB format.
pcolor3	Yes	Third gradient color in hex RGB format.
border	Yes	Single line border - 1 = Yes, 0 = No.
value	No	Predefined (overrides colors) – V1 – V5, H1 – H7, or VU (uses bitmaps below).
imageon	No	Image to use when lit up.
imageoff	No	Image to use when dark.
reverse	Yes	Meter goes backwards (right to left) – only for horz meters.
orient	Yes	horz or vert.

slider

Sliders are used for volume, pitch, etc. You simply provide a bitmap image for the thumb portion of the slider (the part that moves, that you grab with the mouse). The channel (background), is part of the main skin bitmap image.

Attribute	Required	Description/Comments
action	Yes	See table below.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
image	Yes	Bitmap image for the thumb.
tcolor	Yes	Transparent color in hex RGB format.
orient	Yes	horz or vert.

Available actions:

sldxfader	- The crossfader.
x-sldgain	- Volume control for deck x (where x = a,b,c,d).
x-sldpitch	- Pitch control for deck x (where x = a,b,c,d).
x-sldseek	- Seek slider for deck x (where x = a,b,c,d).
x-sldeqlow	- Equalizer low band for deck x (where x = a,b,c,d).
x-sldeqmid	- Equalizer mid band for deck x (where x = a,b,c,d).
x-sldeqhigh	- Equalizer high band for deck x (where x = a,b,c,d).

knob

Knobs can be used in place of sliders for certain operations. Knobs are actually sliders, with a knob type appearance, hence, they are used in a vertical fashion, and not a circular style (in other words, you grab the knob with the mouse, and slide up or down).

Attribute	Required	Description/Comments
action	Yes	See table below.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
imageknob	No	Bitmap image for the knob background.
imagedot	No	Bitmap image for the knob dot that moves around.
tcolor	Yes	Transparent color in hex RGB format.

Available actions:

knb-x-jog	- Jog knob for deck x (where x = a,b,c,d). For seeking forward and backward in a song, stuttering, etc.
knb-x-eqlow	- Equalizer low band for deck x (where x = a,b,c,d).
knb-x-eqmid	- Equalizer mid band for deck x (where x = a,b,c,d).
knb-x-eqhigh	- Equalizer high band for deck x (where x = a,b,c,d).

Notes: The easiest way to do a knob, is to draw it on the main skin background, and leave the **imageknob** parameter empty. Just define the coordinates as the area the knob covers on the background image. If you use a dot image (not required), the smaller the better, as you will get better resolution when using the knob, and Rockit will not have to work as hard painting the dot as it moves.

viewpeak

The viewpeak control, displays the wave form of the song as it plays.

Attribute	Required	Description/Comments
action	Yes	vp-a, vp-b. Currently, only decks a and b have this feature.
visible	Yes	Should we show this on the skin 0 = no, 1 = yes.
x, y, w, h	Yes	Defined in coordinate legend.
linecolor	Yes	Actual lines being drawn color in hex RGB format.
bkcolor	Yes	Background color in hex RGB format.
gridcolor	Yes	Not currently being used.
tcolor	Yes	Not currently being used.

Notes: The larger the viewpeak displays are, the more work Rockit has to do painting them, so for best results, keep them to a minimum size, unless you have a fast processor. There is also a certain amount of overhead in Rockit's decoder when these are visible, as a FFT (Fast Fourier Transform) is done on the samples, which is math intensive.

Button Command Reference Table

Command	Description
about	Brings up about window
help	Opens help file from web in browser
minimize	Minimize
config	Open Configuration settings
close	Close Rockit
mainmenu	Display main menu.
me-file	Display file menu.
me-view	Display view menu.
me-tools	Display tools menu.
me-help	Display help menu.
x-play	Play song
x-stop	Stop song
x-pause	Pause song
x-playpause	Play/Pause toggle
x-restart	Restart song
x-fadein	Start and fadein song
x-fadeout	Fadeout and stop song
x-eject	Brings up the file open dialog
x-previous	Load song at top of history list
x-preset1	Load preset song # 1
x-preset2	Load preset song # 2
x-preset3	Load preset song # 3
x-preset4	Load preset song # 4
x-cue1	Advance to cue point # 1
x-cue2	Advance to cur point # 2
x-loopin	Set loop in start time
x-loopout	Set loop in end time
x-loopstart	Start stop looping
x-eqonoff	Turn EQ on/off
x-eqflat	Reset EQ to flat
x-mute	Mute the output
x-random	Start random continuous play
x-props	Bring up deck properties
x-pitchup	Pitch bend up
x-pitchdown	Pitch bend down
x-eqlowkill	Kill low EQ band
x-eqmidkill	Kill mid EQ band
x-eqhighkill	Kill high EQ band
x-ff	Fast forward
x-rw	Rewind
x-pitchzero	Center pitch
x-viewqueue	View deck x queue in popup window

viewlibrary	View Library/Playlists in My Library tree
viewartist	View artists in My Library tree
viewalbum	View albums in My Library tree
viewgenre	View genres in My Library tree
viewyear	View years in My Library tree
openlibmgr	Open the library manager
openadvsearch	Open the advanced (HDD) search window
viewsearchbox	Open a standard search window
viewsearch	View search in a popup window
searchgo	Start search
vieweffects	View sound effects sampler
viewmixer	View master mixer
viewcomm	View communications server window
viewrequest	View request list in popup window
viewhistory	View history in popup window
viewjukebox	View audio jukebox
viewvideo	View video jukebox
viewcdplayer	View audio CD player

mx-center	Center crossfader
mx-fadeleft	Auto fade left
mx-faderight	Auto fade right
mx-mixleft	Mix to deck A
mx-mixright	Mix to deck B
mx-mixnext	Mix to next available deck
mx-automixon	AutoMix On/Off
mx-automixrandom	AutoMix Random
mx-automixprops	AutoMix Properties
mx-automixq	View AutoMix Queue
mx-faderleft	Move crossfader left
mx-faderright	Move crossfader right
mx-monitora	Toggle monitor output deck A
mx-monitorb	Toggle monitor output deck B

fx-stop	Stop the effects deck.
fx-play	Play current in effects deck.
fx-1	Play sound effect # 1
fx-2	Play sound effect # 2
fx-3	Play sound effect # 3
fx-4	Play sound effect # 4
fx-5	Play sound effect # 5
fx-6	Play sound effect # 6
fx-7	Play sound effect # 7
fx-8	Play sound effect # 8
fx-9	Play sound effect # 9
fx-10	Play sound effect # 10
fx-11	Play sound effect # 11
fx-12	Play sound effect # 12

fx-13	Play sound effect # 13
fx-14	Play sound effect # 14
fx-15	Play sound effect # 15
fx-16	Play sound effect # 16
fx-17	Play sound effect # 17
fx-18	Play sound effect # 18
fx-19	Play sound effect # 19
fx-20	Play sound effect # 20
vj-load	Load a track into the on skin video deck.
vj-stop	Stop track playing in on skin video deck.
vj-play	Play track loaded in on skin video deck.
vj-exmon	Toggle external monitor on/off for on skin video deck.

Text Field Reference Table

Text Field	Description
x-title	Title of song
x-artist	Artist name.
x-album	Album name.
x-genre	Genre.
x-elaptime	Elapsed time.
x-status	Deck status (playing, paused, etc.).
x-bpm	Beats per minute.
x-sps	Samples per second.
x-filetype	Filetype (MP3 or WAV).
x-totaltime	Total time.
x-remaintime	Remaining time.
x-channels	Channels.
x-frequency	Frequency (44100, etc.).
x-pitch	Current pitch percentage up or down.
clock	Current time of day.
uptime	Elapsed time since Rokit started.
txt-assignleft	Crossfader assign left deck letter.
txt-assignright	Crossfader assign right deck letter.

Status Field Reference Table

Status Field	Description
s-automix	Status of the AutoMixer.
s-mycomputer	Currently viewing in MyComputer tree.
s-mylibrary	Currently viewing in MyLibrary tree.
s-tracklist	Main tracklist status.
s-searchlist	Search list status.
s-historylist	History list status.
s-requestlist	Request list status.
s-alist	Deck A queue status.
s-blist	Deck B queue status.
s-mxlist	AutoMix queue status.
s-vidstatus	Video deck on skin status.

Indicator Light Reference Table

Indicator Light	Description
lt-mx-auto	AutoMix on/off.
lt-mx-autornd	AutoMix Random on/off.
lt-mx-monx	Monitor light for deck x (at end).
lt-x-live	Deck x on air light (outputting to mains).
lt-x-preset1	Deck x preset 1 has song.
lt-x-preset2	Deck x preset 2 has song.
lt-x-preset3	Deck x preset 3 has song.
lt-x-preset4	Deck x preset 4 has song.
lt-x-cue1	Deck x has cue 1.
lt-x-cue2	Deck x has cue 2.
lt-x-eqonoff	Deck x eq on/off.
lt-x-pitchcenter	Deck x pitch is centered.
lt-x-fxecho	Deck x echo is active.
lt-x-fxflange	Deck x flange is active.
lt-x-fxrotate	Deck x rotate is active.
lt-x-loopin	Deck x loop in point set.
lt-x-loopout	Deck x loop out point set.
lt-x-loopstart	Deck is looping.
lt-x-mute	Deck x mute on/off.
lt-x-random	Deck x random continuous play is on/off.
lt-x-eqlowkill	Deck x low band eq is killed.
lt-x-eqmidkill	Deck x mid band eq is killed.
lt-x-eqhighkill	Deck x high band eq is killed.