KONTAKT 4 PLAYER

DEVELOPER GUIDE

Developer's Guide written by Daniel Santucci and Adam Hanley

Edited by Patryk Korman

Document Version: 1.1 (01/2010)

Product Version: 4.0 (09/2009)

Special thanks to the Beta Test Team, who were invaluable not just in tracking down bugs, but in making this a better product.

Germany

Native Instruments GmbH
Schlesische Str. 28
D-10997 Berlin
Germany
info@native-instruments.de
www.native-instruments.de

USA

Native Instruments North America, Inc.
5631 Hollywood Boulevard
Los Angeles, CA 90028
USA
sales@native-instruments.com
www.native-instruments.com

NATIVE INSTRUMENTS

Table Of Contents

1 Essential Information to Get You Started

KONTAKT PLAYER Library development does not require a special developer’s version. You can use the release KONTAKT version to begin programming your library. Please contact Native Instruments if you need a version of KONTAKT for development.

! Only if all items (Library and deliverables) have been received by Native Instruments and proven to be correct, the process of creating the software and Master will start.

1.1 Audio Material / Samples

- wav or aiff files with the extension „.wav“ or „.aif“ (all small letters).
- .ncw (native compressed wave) files - for more info on KONTAKT’s compression function please refer to the KONTAKT user manual.
- 44.1, 48 or 96 kHz
- 16 or 24 bit

When creating the samples for the KONTAKT library, please make sure that in the entire library there are no samples which have the same name, otherwise the conversion / encoding of the library can cause problems with assigning the correct samples to the correct instrument patches. This can easily be tested by attempting to copy all of the samples into a single directory.

! It is ESSENTIAL that the sample names have proper extensions. Although OS X does not require them, our Windows-based library encoding utilities will fail on files without proper extensions.

1.2 Naming Conventions

Although virtually any name and length is now acceptable, it makes sense to use caution when naming samples and instruments. Traditionally slashes (\ /) and dots (.) cause errors.

1.3 Organization of Files

In order to make the conversion of the library smooth and easy, please sort all instruments, samples, and presets into four folders.

- Info
- Instruments
- Multis
- Samples

1.3.1 Info Folder

The info folder contains several text files and all pictures of the product. Additionally it can contain any scripts as text files, .png files for your instrument design, and impulse responses. During the encoding process all these items will be packed into a special NKX monolith. This means that the user can't access or change any background pictures and text files. What follows is a list of the items which should be in the info folder upon library delivery.

- Performance view images: Place all performance view images, including backgrounds and knobs in the info folder. All files should be in .png format. Note that when saving an instrument using the "patch and samples" option, the background image will be saved into a Wallpaper's subfolder within the Samples folder. Please move just this image into the info folder.
- LibCopyright.txt: contains copyright information.
- LibDescription.txt: contains library credits and is not length restricted.

- AboutIconLibAuthor.tga (120 x 80 pixels): Library About screen developer logo (deliver with alpha channel)
- AboutIconLibrary.tga (120 x 80 pixels): library logo on About screen (deliver with alpha channel)
- wallpaper.tga (633 x 99 pixels): displayed library browser background also known as the "library box". If using the provided Photoshop templates, please deliver without the Overlay layer.
- InstIcon.tga (33 x 34 pixels): KONTAKT and KONTAKT PLAYER will automatically display this Icon when an instrument of the library is loaded.

Sub-Folders

- Service Center/... for Service Center provide one company logo (company_name.png) with the following requirements: format: png, max size: 100 x 22, other: transparent background. Place this logo into the Service Center sub-folder.
- ir_samples/...
- pictures/...
- scripts/...

! If you have script controlled impulse rsponses, external scripts, or png graphics, these need to be part of the Info folder as well. It is important to place these items into subfolders, whose names should be **ir_samples**, **scripts**, and **pictures** respectively. Note that all letters are lower-case.

1.3.2 Instruments, Multis, and Samples Folders

Within these folders you can use any subfolder structure you want to organize your library. All Subfolders may contain subfolders and sub-subfolders etc. but too many subfolders many slow down the loading process of instruments or multis.

For the Samples folder you MUST use subfolders. Each subfolder should have around a maximum of 1000 files or else browser refresh and folder opening times might be affected adversely.

Keep in mind that the KONTAKT PLAYER Load menus mirror the folder structure you choose.

- Subfolder “Instruments”
    - Subfolder A (containing all instrument files “*.nki”)
    - Subfolder B (containing all instrument files “*.nki”)
    - etc.
- Subfolder “Multis”
    - Subfolder A (containing all multi files “*.nkm”)
- Subfolder “Samples” (containing all subfolders with samples)
    - Subfolder Samples a
    - Subfolder Samples b
    - Subfolder Samples c
    - etc.

Note that it is possible to encode each top level subfolder individually. The advantage is that each subfolder gets its own monolith container. For example, if your library contains bass, guitar and drum instruments, you could organize your sample subfolders into three categories, i.e. Bass Samples, Guitar Samples, and Drums Samples respectively.

1.3.3 Impulse Responses

You may want to include impulse responses with your library. Note that the impulses which come with KONTAKT are not available for commercial use with your library. When you include impulses you have the choice to either protect them inside the monolith container or not.

If you wish to protect Impulse response samples, then put them in a folder called **ir_samples** inside the Info folder.

1.3.4 Resaving Instruments with the New Library Structure

KONTAKT contains a unique batch save process which can be found under the File/Load menu. When all files are sorted the above way, please “batch resave” all instruments and multis with the option “relative path” (i.e. deselect absolute path in the save dialog!).

1. Choose batch resave from the file/load menu.
2. Select the folder which should get resaved, in this case it is the instrument and multis folders.
3. Point the samples missing dialog to the newly organized samples folder and choose save.

> ! This process will take some time. Please allow saving to work uninterrupted. You‘ll know saving is done when the samples missing dialog has closed.

Afterwards, try to load all instruments and multis from their new location to make sure they load properly and find all samples.

1.4 Library Information (Including Instrument Options)

1.4.1 Documentation Folder

The documentation folder is used for any library relevant information (i.e. Documentation (.txt and PDF), web links, video tutorials, and basically anything else you can think of. The contents of this folder will be listed under the library information (i) button displayed in the library box.

Please deliver all documentation in acceptable cross-platform formats. That means any text documents should be .txt or PDF.

- Save all web links in .webloc or .url format. In most browsers this is done by dragging the url to a location outside of the window to save it. Then place this file into the Documentation folder.

1.4.2 Readme/ Welcome PDF

Please deliver a readme (preferably PDF or .txt) that belongs to the library. This readme can be altered to suit your needs. It can be decided to also have both a Welcome PDF, as in a getting started guide, and a text file readme. In this case, it is recommended to put any last minute changes into the readme and use the Welcome PDF as a quick start. Items of the readme that might/should be exchanged or updated:

- Product logo
- Packshot
- License agreement
- Credits
- Readme texts
- Information about the library
- Title graphics
- URLs

All changes will be performed by the OEM partner.

1.4.3 Documentation

Native Instruments will provide the KONTAKT PLAYER documentation in form of a Getting Started guide plus the KONTAKT manual. You are however responsible for your library's documentation. No product will be released without appropriate documentation. You will be provided a documentation template and a documentation style guide to guide you through the documentation process. Please contact Native Instruments for further information.

1.5 GUI

Sample developer's logo in the Library box format as described in the Info folder section, wallpaper.tga.

1.6 Library Name

The Library name will be used on many occasions:

- the preferences
    - com.native-instruments. [product name].plist
    - [product name].plist
- the registry
- the installer
    - [product name]_Setup PC.exe
    - [product name]_Installer Mac.mpkg
- the CD/DVD name
- Service Center

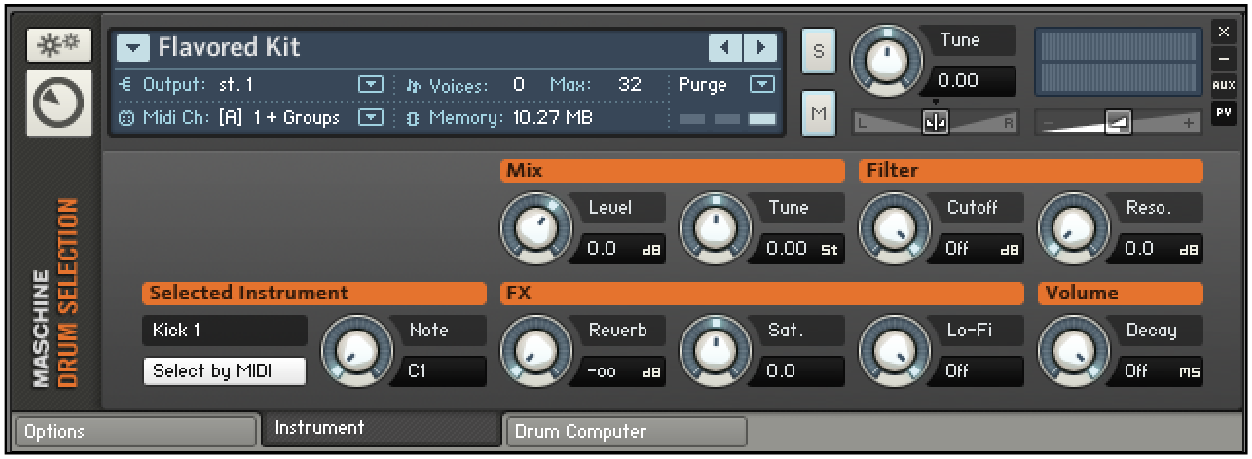
The library name should not be longer than 27 characters.

1.7 Delivery Medium

Please either use a DVD-R (single or dual), CD-R or Hard Drive for delivery of the Library (depending on the size of the library). Burn the CD/DVD as Macintosh and PC hybrid compatible.

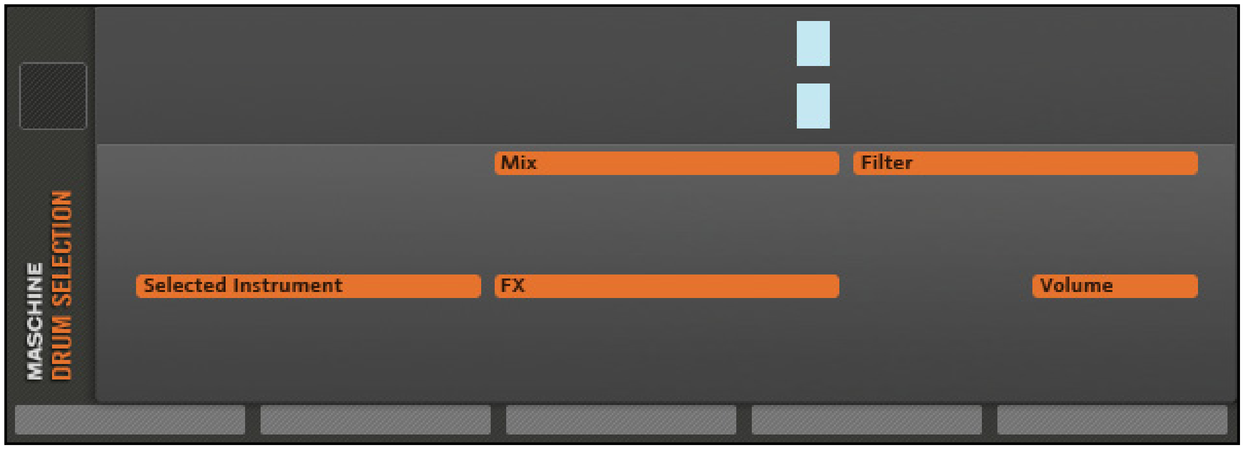
2 KONTAKT PLAYER Basics

2.1 Performance View

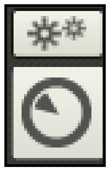

Flavored Kit
Output: st. 1
Voices: 0 Max: 32
Midi Ch: [A] 1 + Groups
Memory: 10.27 MB
Tune 0.00
Level 0.0 dB
Tune 0.00 st
Cutoff Off dB
Reso. 0.0 dB
Kick 1
Note C1
Reverb -oo dB
Sat. 0.0
Lo-Fi Off
Decay Off ms

The Performance View is a customizable "easy-edit" page for each instrument. With this view you can design the look and control elements of your interface for each instrument of your library individually. Up to 5 tabs are possible. Each tab can have its own layout. In the above performance view the tga image contains the labels and background surface, including the MASCHINE Drum Selection text.

Here is how a performance view tga looks for all tabs without any control elements:

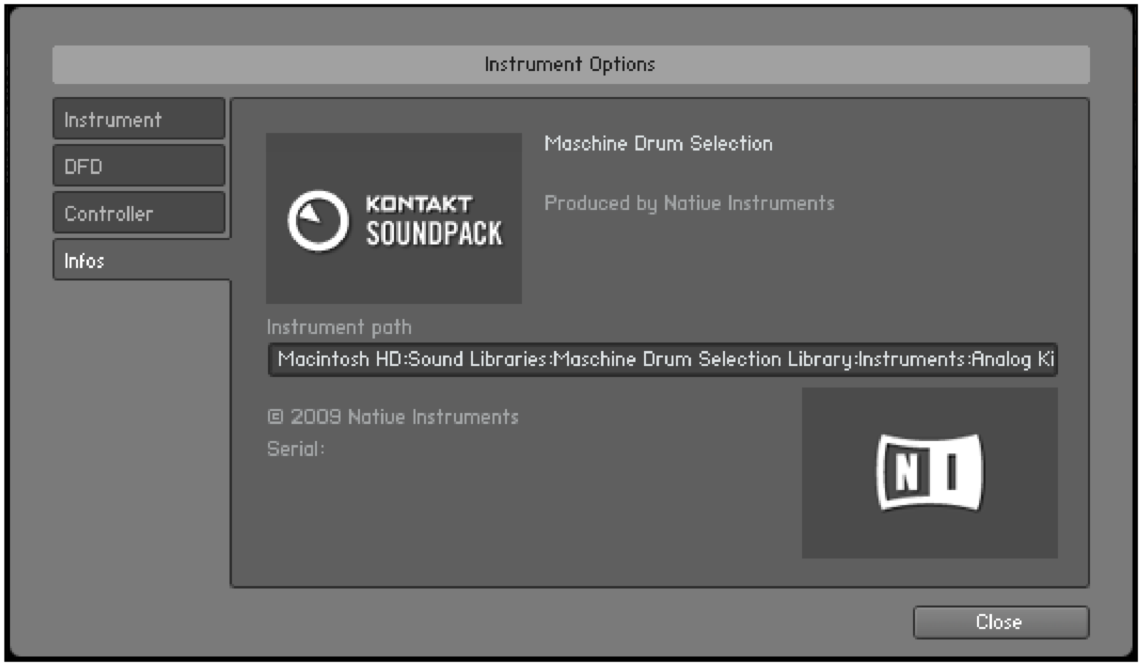


You can unfold the performance view by clicking the Instrument Icon (InstIcon.tga) under the gear symbol.



For KONTAKT PLAYER functionality, the gear symbol replaces the KONTAKT “edit page” wrench button. This symbol means that the instrument cannot be opened for editing. In other words it is locked. If an instrument is “locked” during encoding, access to the edit page , even in the full version of KONTAKT, is also prohibited. The library developer can decide if the instruments should be locked or not. A click on the gear symbol opens the **instrument options** screen.

2.2 Instrument Options Screen



Maschine Drum Selection

Produced by Native Instruments

Instrument path: Macintosh HD:Sound Libraries:Maschine Drum Selection Library:Instruments:Analog Ki

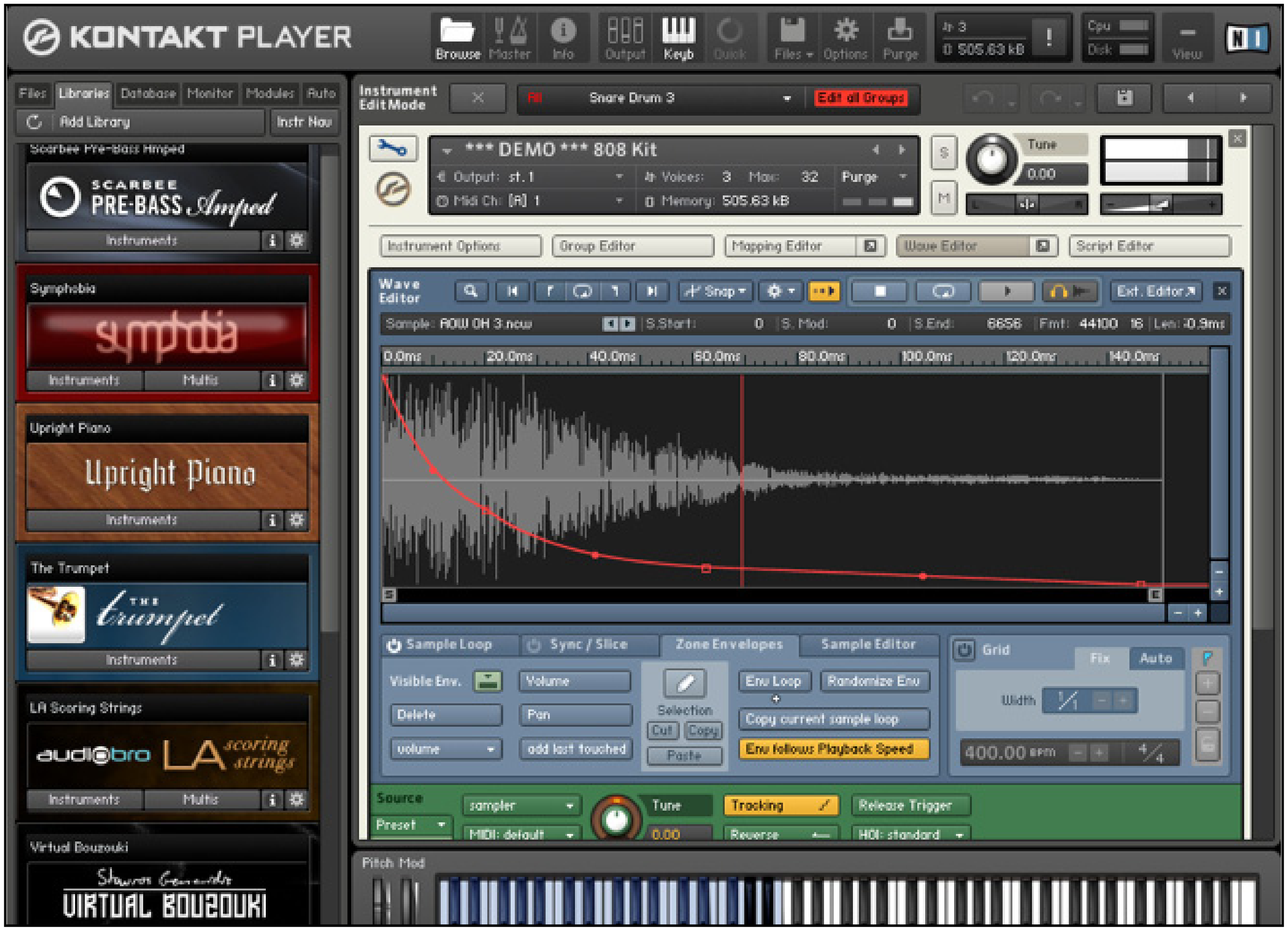
© 2009 Native Instruments

Serial:

Each library has its own Info screen where, in addition to the instrument specific options, all related library information like credits, copyright, serial number, and Instrument installation path are displayed. This screen is also where the two Info graphics AboutIconLibAuthor.tga and AboutIconLib.tga are displayed.

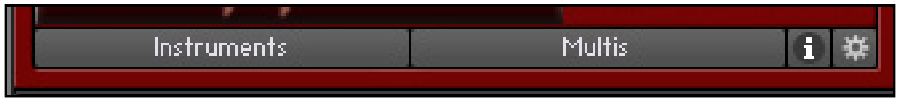
2.3 The Library Browser

The library browser is on the left side of the player and shows one box for each installed library.

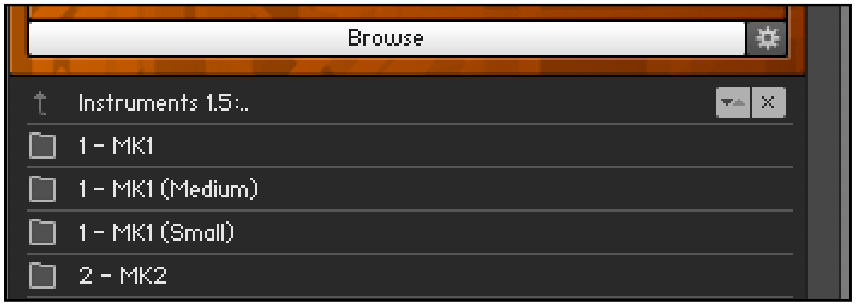


Loading Instruments from the Library Browser

The user can access the instruments of all installed libraries by clicking the Instruments or Multis buttons of the corresponding library.



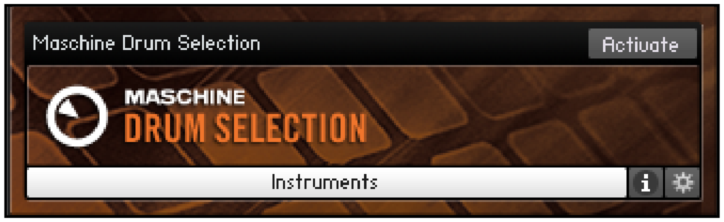
There is one way of browsing the instruments of each library. A click on the instrument button in the library box will switch to the standard KONTAKT browser view with a folder and file list.



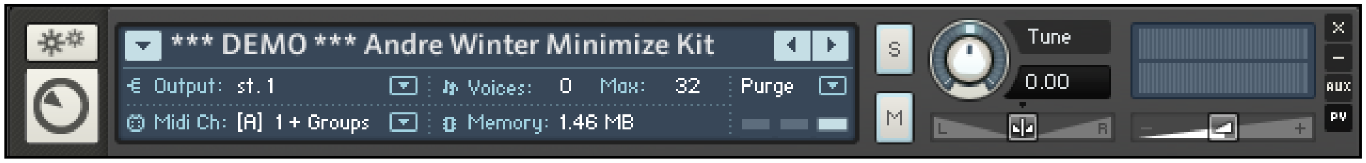
You can use the back arrow to navigate one level higher or use the x button to close the library box. The directional arrows will resort the folders or instruments from last to first.

2.4 Registration and Authorization in Detail

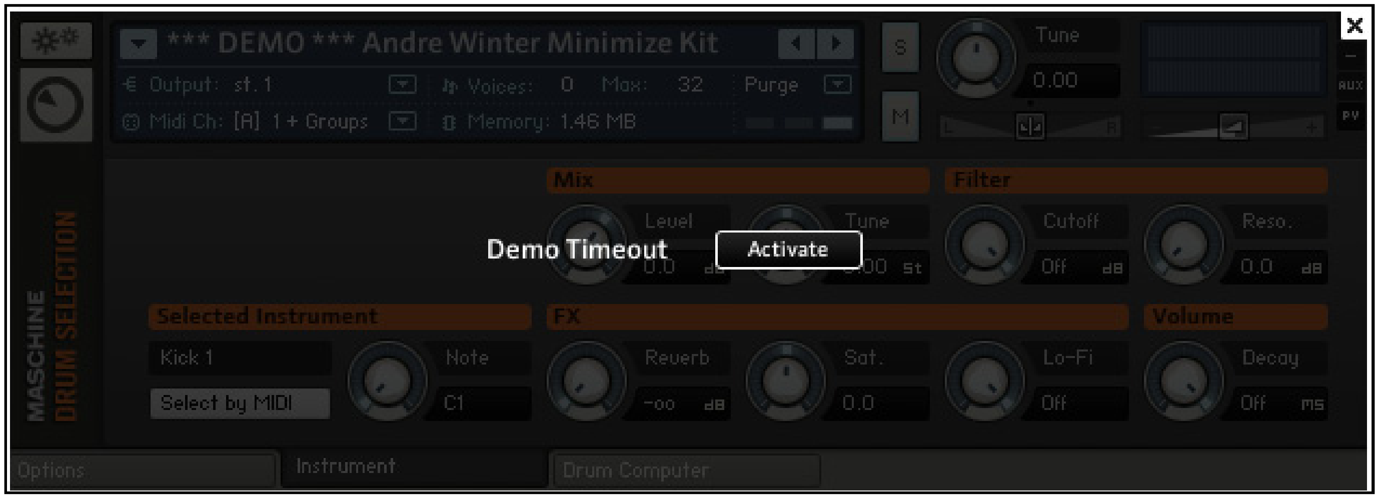
After installation the library runs in 15 minute demo mode. You can recognize that the library is in demo mode if the **Activate** button appears in the Library Box.



Furthermore, when in demo mode instruments will be loaded with the title “***DEMO***” in the name field.



The Demo Mode allows you to play any Instruments from a particular library for 15 minutes. The countdown begins when the first Instrument from that given library is loaded. After 15 minutes have passed there will be a "Demo Timeout" message and any other Instruments loaded from that library will also be timed out.



2.5 Library Encoding Options

KONTAKT PLAYER provides you with different options concerning the copy protection process.

Standard KONTAKT PLAYER Library

The library is delivered with the KONTAKT PLAYER engine and can be used directly out of the box. If the user already owns a KONTAKT PLAYER library, an additional library box will appear in the player browser after a successful installation. The user can mix instruments of all installed libraries in one KONTAKT PLAYER instance.

KONTAKT PLAYER libraries have the following protection.

- NKIs, NKMs, NKBs of the copy protected product can only be loaded into KONTAKT PLAYER or KONTAKT if the library is authorized or still in demo mode.
- Samples and graphic content of the library will be packed in encrypted NKX files.

“Lock” Option

KONTAKT PLAYER libraries can be “locked.” Meaning that even in KONTAKT the user has no access to the edit page or the samples in NKX files. If you want to protect the programming work and group management concept you developed in KONTAKT, this is the encoding option with the highest security. Please keep in mind that the user cannot create own instruments in this library option. The user can only edit the provided performance view features provided and save the instruments.

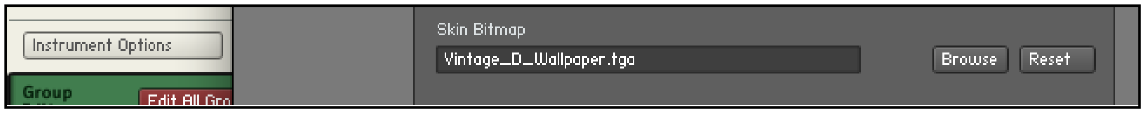
The **gear icon** indicates the “locked” status of the library in KONTAKT. KONTAKT PLAYER always shows the gear tool because the edit view is not available for any instrument.

3 Library Structure and File Components

KONTAKT PLAYER instruments are created in the latetst KONTAKT release version.

The instrument consists of the samples, the mapping and group settings, the user interface script and the background picture.

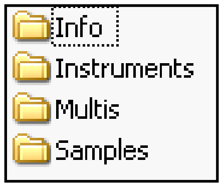
The background picture is chosen in the instrument options on tab 1, Instrument.



Skin Bitmap: Vintage_D_Wallpaper.tga

The background skin can be in TGA or PNG file format. PNG are generally smaller, so we recommend using them. Please use one of our Photoshop templates (e.g. instrument_1row_flat.psd) in the library development kit to create your library skin.

The library has to be delivered in a fixed file structure containing four subfolders.

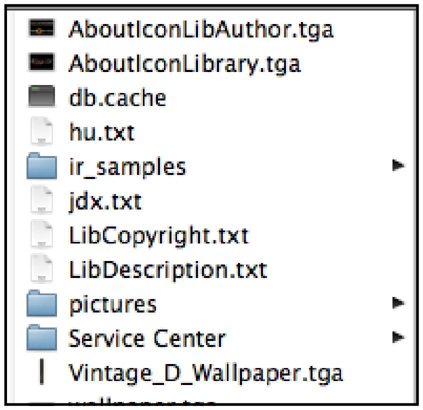


Info
Instruments
Multis
Samples

Instruments, Multis and Samples folders should be self explanatory. Within these folders you can use any subfolder structure you want to organize your library. Keep in mind that the KONTAKT browser mirrors the folder structure you choose.

Info Folder

The info folder contains several text files and all pictures of the product, including instrument backgrounds. During the encoding process these will be packed into a special NKX monolith which means that the user can't access or change any background pictures and text files.



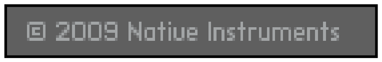
AboutIconLibAuthor.tga
AboutIconLibrary.tga
db.cache
hu.txt
ir_samples ▸
jdx.txt
LibCopyright.txt
LibDescription.txt
pictures ▸
Service Center ▸
Vintage_D_Wallpaper.tga

The text files contain the Instrument Information e.g. library credits, and copyright.

All text files have to be Unicode formatted. Mac users might have to reformat the txt files. OS X's standard text editor (TextEdit) can handle Unicode texts.

- **LibCopyright.txt**

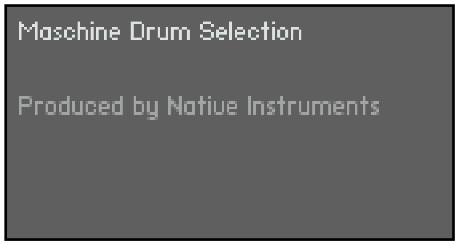
  Contains Copyright text and is restricted to 45 characters.

  © 2009 Native Instruments

- **LibDescription.txt**

  Contains library credits and is not length restricted.

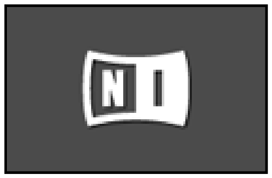
  Maschine Drum Selection

  Produced by Native Instruments

  The description text will show a scroll bar on the right side if necessary. Line breaks are possible.

- **AboutIconLibAuthor.tga**

  Library About screen developer logo (deliver with alpha channel)

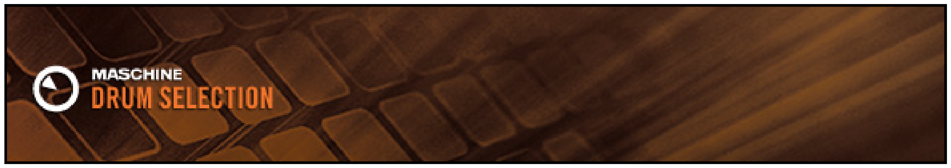
- **AboutIconLibrary.tga**

  Library logo on About screen (deliver with alpha channel)

  KONTAKT
SOUNDPACK

- **wallpaper.tga**

  Displayed library browser background. Note that the KONTAKT is expandable, small, mid, and large. Therefore please make sure that the wallpaper is designed to accommodate this variability. If using our templates, please deliver without the overlay layer.

- **InstIcon.tga**

  KONTAKT and KONTAKT PLAYER will automatically display this Icon when an instrument of the library is loaded.

4 Conception of KONTAKT PLAYER Libraries From Scratch

KONTAKT PLAYER libraries are not just sampler patches, they are optimized user interfaces as well. We recommend spending a bit of time on the overall concept of your library/instruments. It's possible to create one general interface for an entire library, but this is not always the most elegant solution. String patches might need a totally different interface than a Piano. Therefore, it might make sense to categorize your library and create dedicated interfaces.

Highly customized GUIs require additional work on concept and scripting, but in the end this makes the library much more valuable. To get some ideas of what is possible, we recommend that you check out the KONTAKT factory library. All KONTAKT instruments are intentionally unlocked, as we encourage you to explore (and in some cases even use) our programming.

! Bear in mind that there are some restrictions as to which factory scripts can be used commercially, so if you want to use anything in your product, please be sure to ask for our permission first.

If you don't feel comfortable with making your own scripts, or would like something very unique, please get in touch with us. We know many freelance programmers which may be able to help you realize your vision.

4.1 Scripting

For in depth script information please consult the KONTAKT Documentation folder. There are comprehensive KSP documents, which include all the latest features and additions. There are also comprehensive online tutorials, especially noteworthy is one from Nils Liberg (http://nilsliberg.se/ksp/scripts/tutorial/).

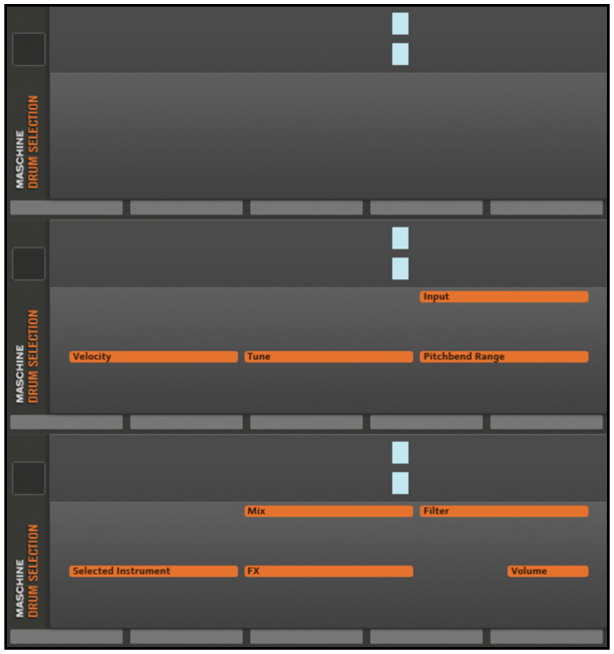
Almost every parameter on the full KONTAKT edit page can be accessed from script and implemented in the performance view. When designing your interfaces, keep in mind that KONTAKT PLAYER users only have access to the controls which you declare in the performance view. So careful attention should be given to which engine parameters you make available. While this document is meant to serve as an introduction to KSP, you should have a rough overview of basic scripting and declaring GUI elements to understand the following guide.

4.2 Enabling the Performance View

The call “make _ perfview” enables the performance view for the instrument. You can use the command in any or all of the five script slots in the “init” callback. Keep in mind that MIDI processing can be done in the remaining 4 slots where the signal flow goes from slot 1 thru to slot 5, meaning that MIDI events generated in slot 1 can be used to transfer information to scripts in following slots.

4.3 Performance View Tabs

It is possible to have up to 5 performance view tabs per instrument. Furthermore, each tab can have a different knob layout or graphical elements. Below you see a background tga for a 3 tabbed instrument.



use the make _ perfview command in each tab to activate performance views.

To address the different backgrounds use the command:

`_ set _ skin _ offset (offset in pixels)`

whereby offset can also be done with an operator as such (0 * 255), (1 * 255) and so on.

Customizing the Performance View

There are a few options which are useful for customizing your library's user interface controls, thus making it look even more original. This is accomplished by way of hiding KSP's standard GUI elements and in turn incorporating them into your background tga files.

- The elements whichbackground of knobs, labels, value edits, and tables
- value of knobs
- title of knobs
- mod ring light of knobs

The command necessary for this is:

`hide _ part(<control-variable>,<hide-mask>)` wherein `<hide-mask>` is a bit by bit combination of the following constants:

- `$HIDE _ PART _ BG`
- `$HIDE _ PART _ VALUE`
- `$HIDE _ PART _ TITLE`
- `$HIDE _ PART _ MOD _ LIGHT`

The first mentioned constant "`$HIDE _ PART _ BG`" can be used on knobs, labels, value edits and tables.

"`$HIDE _ PART _ VALUE`," "`$HIDE _ PART _ TITLE`" and "`$HIDE _ MOD _ LIGHT`" can only be used on knobs.

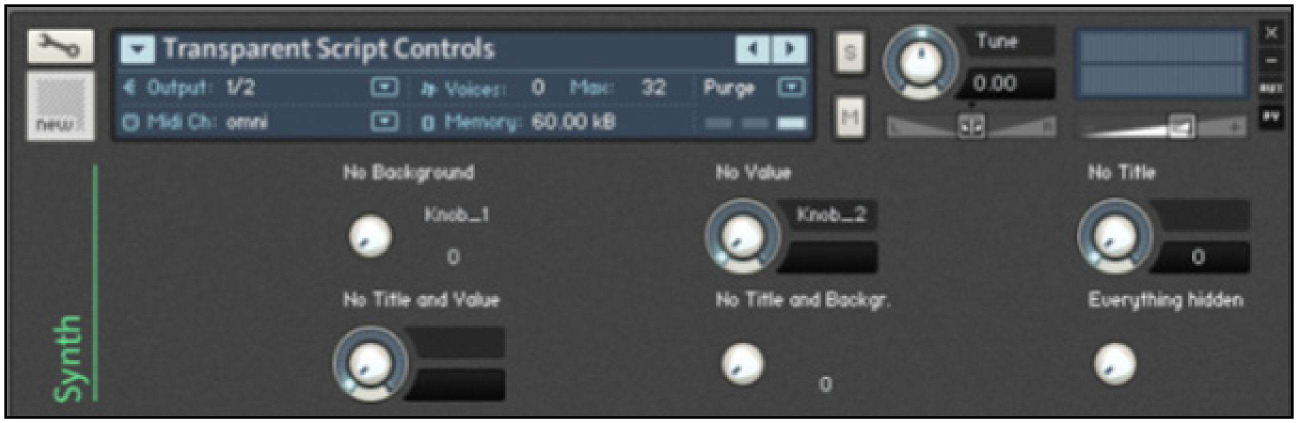
If you want to hide a knob's background and title, but keep its value, you would use:

```
on init

  make _ perfview
  declare ui _ knob $Knob (0,100,1)
  hide _ part($Knob,$HIDE _ PART _ BG .or. $HIDE _ PART _ TITLE)

end on
```

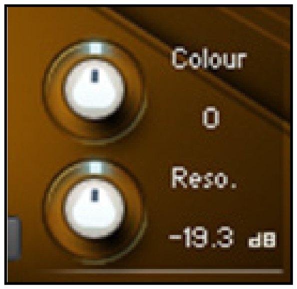
! Please note the bit by bit 'or' operator (`.or.`) in this example. Without the dots before and after 'or' it would not be working because it would be recognized as logical operator by the KSP.

! Please note: `hide _ part()` is only available in the init callback.

Here's a simple example of the various combinations:



This example shows a knob ring designed into the background tga and hidden values and titles:



5 New Performance View Possibilities in KONTAKT 4

This guide introduces some of the new Performance View options made available by recent updates to KONTAKT's script language. It will cover how to use most of the new commands as well as offering some examples of them in practice. It is possible to still use the older performance view options offered by earlier versions of KONTAKT, which may be preferable if you do not have access to good instrument interface design resources and need a simple way to produce instrument controls.

5.1 Editing UI Elements by Pixel

In KONTAKT 4, it is now possible to edit and move user interface elements in pixels, rather than by the previous grid system. Most of the UI commands for moving and resizing have been updated with pixel versions that mostly follow the rule of adding `_ px` to the end of the older commands (check the KSP reference manual for a full list of new pixel based commands).

Example: Positioning a UI element:

1 by grid: `move _ control(<ui _ name>, <x,y coordinates by grid>)`

2 by pixel: `move _ control _ px(<ui _ name>, <x,y coordinates by pixel>)`

It is important to note that there can be issues when using combinations of grid commands and pixel commands. In our experience it is better to choose one format at the start and stick with it.

5.1.1 Overlaying UI Elements

By using the pixel placement method, it is now possible to overlay UI elements. All elements are generated in order of type, so the element layer priority is automatically set by KONTAKT's engine and there is no way to over-ride that. This will be further explained in the tutorial.

! Avoid overlapping of controls which have mouse interaction (ui_knob, ui_slider, ui_switch, ui_button,_ui_table, ui_value_edit).

5.2 New User Interface Controls

In the KONTAKT 4 script update, two new controls have been introduced: Sliders and switches.

Sliders act as you would expect, but they can also be used to replace the old KONTAKT knobs (as described in section 5.3.4). A slider can have a custom skin applied to it, but a knob (being a more complex element) cannot.

Switches are virtually identical to buttons, but have the added benefit of being automatable. The button has not been replaced as it is still useful in certain cases, for example, switching between control views (a function that does not need automating).

5.3 New User Interface Commands

5.3.1 UI ID

With the recent update to the KONTAKT script features, it is now possible to edit user interface elements in many ways. Almost all of the new commands require addressing the element by it's ID number (these are automatically assigned during initialization).

► The UI ID can be retrieved by using the `get _ ui _ id(<ui element name>)` command.

The UI IDs are generated each time the script is initialized, so it is always advised to use the above command when addressing a UI element by its ID number.

5.3.2 UI Control Parameters

Most attributes for controls are edited using the new `set _ control _ par()` or `set _ control _ par _ str()` commands. For skinning UI elements, you should familiarize yourself with these commands and their variables in the KSP reference manual.

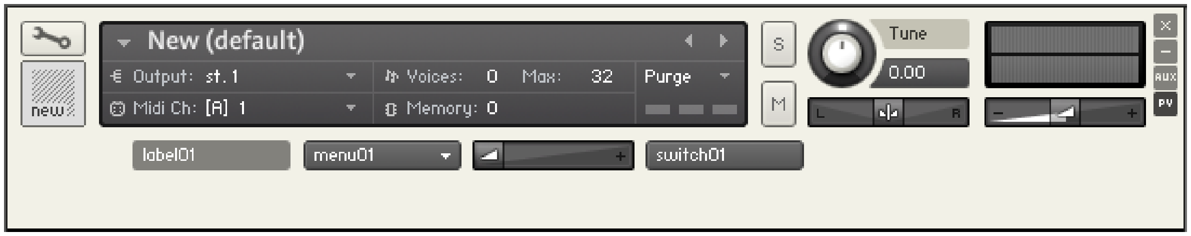
5.3.3 Skinning UI Elements

It is now possible to skin (e.g., apply a custom image file) to various control elements. The following is a short scripting tutorial to show you how to skin and adjust controls using the new commands.

► Enter the following script into the KONTAKT script editor:

```
on init

{--Set performance view attributes--}
   make_perfview
   set_ui_height_px(50)
   set_script_title("Example Script")

{--Declare UI elements--}
   declare ui_slider $slider01 (0,100)
   declare ui_switch $switch01
   declare ui_label $label01 (1,1)
   declare ui_menu $menu01


end on
```

In the above script we have made a performance view called "Example Script", which is 50 pixels high and includes 4 different UI elements. The instrument should look like this:



! Note that the height of the performance view does not include the KONTAKT Instrument main controls, so instruments are always a little bigger than the set height—an important fact when making custom wallpapers.

Also note the order of the controls. In the script the slider is declared first, but on the interface the label appears first. This is because of the UI element generation order as described on the previous page. The order of the elements as they appear here is the order of element layer priority from lowest to highest (so, for example, a slider will always be placed above a label, if they are moved to the same space).

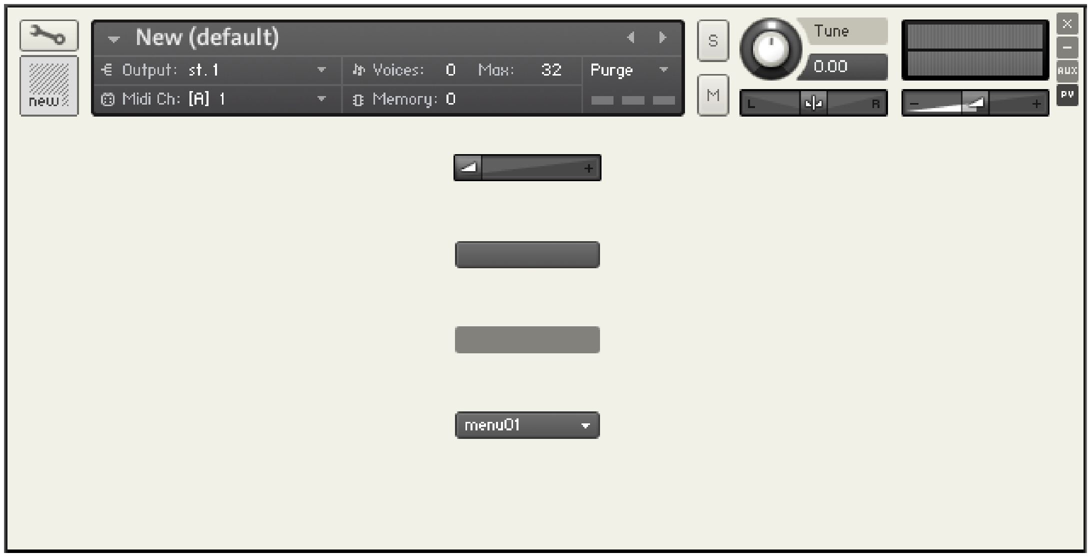
Let us adjust these controls using some of the new commands.

In the following example, the UI IDs will be stored in an array, which will later be accessed using a loop counter. This is a useful technique that is made possible with the new `set_control_par()` commands.

► Copy and paste the following script into KONTAKT's script editor:

```
on init

{--Set performance view attributes--}
    make_perfview
    set_ui_height_px(250)
    set_script_title("Example Script")

{--Declare UI elements--}
    declare ui_slider $slider01 (0,100)
    declare ui_switch $switch01
    declare ui_label $label01 (1,1)
    declare ui_menu $menu01

{--Place UI IDs in an Array--}
    declare %ui_id_array[4]
    %ui_id_array[0] := get_ui_id($slider01)
    %ui_id_array[1] := get_ui_id($switch01)
    %ui_id_array[2] := get_ui_id($label01)
    %ui_id_array[3] := get_ui_id($menu01)

declare $count := 0

    while ($count < 4)
        set_control_par(%ui_id_array[$count],$CONTROL_PAR_POS_X,262)
        set_control_par(%ui_id_array[$count],$CONTROL_PAR_POS_Y,18+($count*50))
        set_control_par_str(%ui_id_array[$count],$CONTROL_PAR_TEXT,"")
        inc($count)
    end while

end on
```

→ This will produce a performance view that looks like this:



Here, we have re-arranged the controls in a vertical grid by addressing the elements by their UI IDs in a while loop, meaning that we can apply the same processes to all of these elements. This is very useful when skinning multiple controls, for example.

Now we will skin the UI elements.

► First of all, place the provided image files (and their corresponding text files) in the KONTAKT 4 user pictures folder, which can be found here:

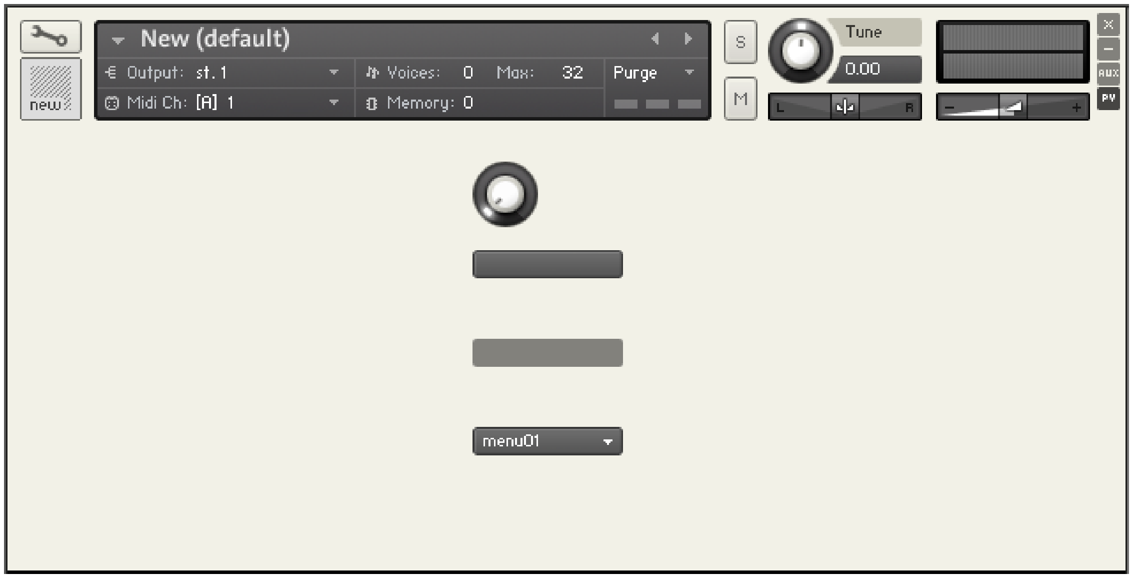
- On Windows XP: *Documents and Settings/User/My Documents/Native Instruments/Kontakt 4/Pictures*
- On Mac OSX or Windows Vista / Windows 7: *User/Documents/Native Instruments/Kontakt 4/Pictures*

When you reference an image file in the KONTAKT script, it automatically checks the library pictures folder, then the user pictures folder. This is why the image file path does not need referenced. Also, the image file extension (.png) does not need to be included in the script either.

► Now add the following text to the script we were already using (before the *end on* command):

```
set _ control _ par _ str(get _ ui _ id($slider01),$CONTROL _ PAR _ PICTURE,"exins _ poti _ 129")
```

► Click apply and you should have something that looks like this:



We can see here that the slider has now become a knob. In KONTAKT 4 it is important to not view a slider as only a horizontal slider, but as a skinnable object that acts like a slider (ie. is controlled by click and drag). By adding the image of a knob, we can create custom knobs that do not have to look like the default KONTAKT knobs.

However, by skinning a horizontally controlled slider with a vertically controlled knob, we introduce a control problem. We need to tell KONTAKT to change to control direction of the slider, so that the new knob acts as expected.

► Add this line of code to the script:

```
set_control_par(get_ui_id($slider01),$CONTROL_PAR_MOUSE_BEHAVIOUR,-500)
```

The value of -500 controls the response of the mouse. In this command, negative values represent vertical control and positive values horizontal control. The higher the numeric value, the faster the response will be. The strength of these numbers does relate to the size of the control, so for each control type it is best to adjust the number through testing, though a value of +/- 500 is usually a good starting point.

5.3.4 Changing the Instrument Icon

As well as being able to skin certain UI elements, it is now also possible to apply a custom image to the instrument icon (the small graphic to the top left of every instrument) or even just hide it completely. You can access the parameters for the instrument icon still by using the `set_control_par()` and `set_control_par_str()` commands, using the constant `$INST_ICON_ID` for the UI ID.

Here are the lines of code you will need to customize the instrument icons:

- To apply a custom image:
- `set_control_par_str($INST_ICON_ID,$CONTROL_PAR_PICTURE,"<image name>")`
- To hide the icon: `set_control_par($INST_ICON_ID,$CONTROL_PAR_HIDE,0)`

5.4 Preparing Image Files

5.4.1 Accompanying Text File

Each image must be accompanied by a text (.txt) file of the same name containing important information on how KONTAKT should read this image. If a text file is not provided, any image file in the user picture folder will have a default one generated. If the image file is in the factory folder a text file will not be generated, but the image will be read by KONTAKT as if the values were set to default.

Here is an example of the content of one of these text files:

```
Has Alpha Channel: yes
Number of Animations: 6
Horizontal Animation:  no
Vertical Resizable: no
Horizontal Resizable: yes
Fixed Top: 0
Fixed Bottom: 0
Fixed Left: 0
Fixed Right:  17
```

What this means:

1 `Has Alpha Channel`: (yes or no) tells KONTAKT whether or not the image file contains a transparent (alpha) layer. If in doubt, set this to “yes”, as this will cause no problems either way.

2 `Number of animations`: sets the number of frames in the image. KONTAKT then equally divides the image using this number.

3 `Horizontal Animation`: (yes or no) designates in what direction the image should be split.

4 `Vertical Resizable`: (yes or no) if set to yes, it is possible to adjust the height of the image file in KONTAKT using the $CONTROL_PAR_HEIGHT constant in the set_control_par command.

5 `Horizontal Resizable`: (yes or no) if set to yes, it is possible to adjust the width of the image file in KONTAKT using the using the $CONTROL_PAR_WIDTH constant.

6 `Fixed Top/Bottom/Left/Right`: if you wish to be able to resize an image, but keep a certain area around the edge intact, simply set the number of pixels in the desired area. In the above example, the image is able to be resized horizontally, but the 17 pixels on the right side of the image will not be stretched.

It is important to note that KONTAKT is currently very sensitive about text file line break encoding. In our experience, the safest software to use is Notepad on Windows. This is unfortunately also true for scripts that link to external text files.

5.4.2 Animations

Image files to be used in KONTAKT must be saved as .png files.

In order for KONTAKT to treat the image correctly, it must be provided in as stack of animations. So if you want one knob with 100 states, you need to build a very long image with all of these states in a row or column. KONTAKT will read the file top to bottom or left to right, depending on the setting of the .txt file. So your lowest value for a knob should be at the top or left and the highest at the bottom or right.

Switches and Dropdowns should always have six animations, in a set order:

1. off
2. on
3. off, mouse down
4. on, mouse down
5. off, mouse over
6. on, mouse over

The following script will illustrate the different states of a switch.

► First, copy the image and text files “switch_state_example” to the KONTAKT 4 user pictures folder. Then copy and paste the following script into a new KONTAKT instrument:

```
on init

   make _ perfview

   declare ui _ switch $switch _ states
   set _ control _ par _ str(get _ ui _ id($switch _ states),$CONTROL _ PAR _ TEXT,"")
   set _ control _ par _ str(get _ ui _ id($switch _ states),$CONTROL _ PAR _ PICTURE,"switch _ ...
state _ example")

end on
```

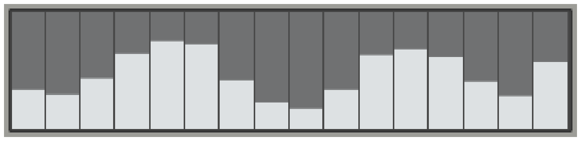
This will produce a single button that changes color depending on its state:

1	off	black
2	on	red
3	off, mouse down	orange
4	on, mouse down	yellow
5	off, mouse over	purple
6	on, mouse over	green

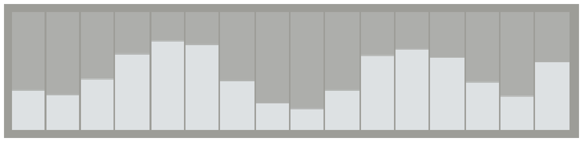
Image files must always have six animations, even if you only plan to use the basic on/off. In this case, just triplicate the two on/off animations to bring the total up to six.

5.4.3 Tables

It is also possible to edit the appearance of tables in KONTAKT 4, but there are some behaviors to take note of. Hiding the background of a table will remove a label from behind it, but the semi-transparent bars of the table will remain. This is illustrated here:

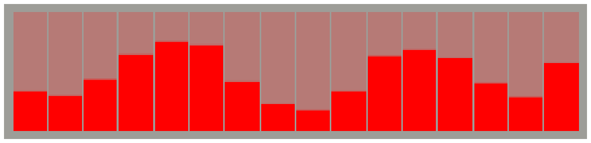


Example of a table with background.



Example of the same table with the background hidden.

Changing the color of the bars using the constant `$CONTROL_PAR_BAR_COLOR` requires a hex value to specify the color. The hexadecimal value needs a "9" placed at the start and an "h" placed at the end in order for KONTAKT to read it as a hexadecimal number and not a string. This will give you an 8 digit color code reading like "9FF0000h" (which represents red). Changing the color of the bars also changes the color of the semi-transparent background:



5.4.4 Fonts

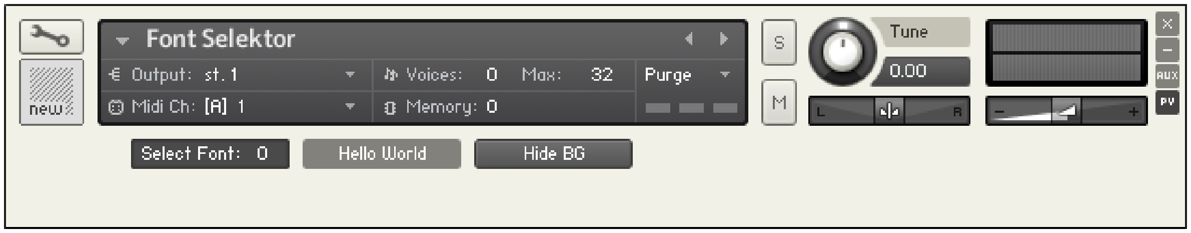
KONTAKT 4 now has a collection of 25 font variations to choose from for any UI element that uses text. These fonts are hard coded into the program, so you cannot use any custom fonts outside of this collection. The fonts are selected using the `$CONTROL_PAR_FONT_TYPE` constant in the `set_control_par()` command, followed by a vale from 0 – 24.

The text alignment can also now be adjusted using the `$CONTROL_PAR_TEXT_ALIGNMENT` constant, followed by a value from 0 – 2. The following is a key that describes which alignment corresponds to each value: 0 = left, 1= center, 2 = right.

► Copy and paste the following script into KONTAKT 4:

```
on init

    make_perfview

    {--Create and adjust example label--}

    declare ui_label $label_1 (1,1)
    set_text($label_1,"Hello World")
    set_control_par(get_ui_id($label_1),$CONTROL_PAR_TEXT_ALIGNMENT,1)
    move_control($label_1,2,1)

    {--Create and adjust value edit to control font type of label--}

    declare ui_value_edit $font_select (0,24,1)
    set_text($font_select,"Select Font")
    move_control($font_select,1,1)

    {--create and adjust button for hiding the label background--}

    declare ui_button $hide_bg
    set_text($hide_bg,"Hide BG ")
    set_control_par(get_ui_id($hide_bg),$CONTROL_PAR_TEXT_ALIGNMENT,1)

end on

{--value edit callback for changing label font--}

on ui_control($font_select)
    set_control_par(get_ui_id($label_1),$CONTROL_PAR_FONT_TYPE,$font_select)
end on

{--button callback for hiding label background--}

on ui_control ($hide_bg)
    if($hide_bg = 0)
        set_control_par(get_ui_id($label_1),$CONTROL_PAR_HIDE,$HIDE_PART_NOTHING)
    else
        set_control_par(get_ui_id($label_1),$CONTROL_PAR_HIDE,$HIDE_PART_BG)
    end if
end on

{--end of script--}
```

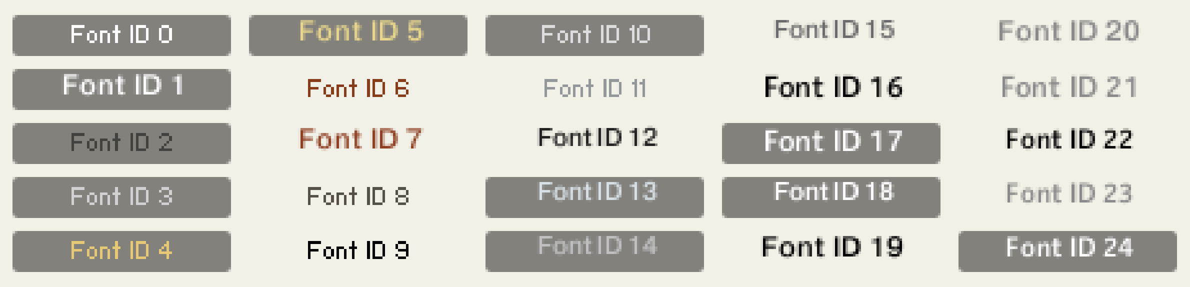
→ This should produce something that looks like this:



With this we have created a KONTAKT font reference. You can select a font using the value edit, the font will then be displayed in the "Hello World" label. Since some of the fonts can be difficult to see against the background of the label, a button is included to hide this background if necessary. The text of the label and button have been center-aligned to produce a neater interface.

As you cycle through the fonts, you may notice that some do not have a centered vertical alignment. It is possible to adjust this now with the `$CONTROL _ PAR _ TEXTPOS _ Y` constant in the `set _ control _ par()` command. The value you enter for this command will add a Y position offset from the top of the element in pixels.

For reference, here is a guide with all 25 font variations:



Font ID 0	Font ID 5	Font ID 10	Font ID 15	Font ID 20
Font ID 1	Font ID 6	Font ID 11	Font ID 16	Font ID 21
Font ID 2	Font ID 7	Font ID 12	Font ID 17	Font ID 22
Font ID 3	Font ID 8	Font ID 13	Font ID 18	Font ID 23
Font ID 4	Font ID 9	Font ID 14	Font ID 19	Font ID 24