MaXX Settings Configuration Management

Instrumentation Guide

Version 0.22

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Versions

Version	Date	Author(s)	Description
0.10	2020-12-30	Eric Masson	Initial and ongoing work
0.20	2021-01-06	Eric Masson	Complete documentation separation from Technical Specification.
0.21	2021-01-13	Eric Masson	Start adding Implementation details for Instruments, added SgiScheme Choice detail.
0.22	2021-04-10	Eric Masson	Sync with Technical specs v. 0.98

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Table of Content

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Synopsys

MaXX Settings is a dynamic configuration management subsystem designed from the ground up with simplicity in mind while not sacrificing flexibility and extensibility. MaXX Settings comes with its own CLI interface allowing simple management, automation via scripting, inline-query and easy application integration. MaXX Settings also provides Java and C++ binding making it super easy to integrate within most modern applications. MaXX Settings allow the definition of System wide setting, we call them **Instruments**, and user's overridables called **User Preferences**.

This document will dive into MaXX Settings implementation details and the MaXX Interactive Desktop Instrumentation.

The reader is expected to have read the MaXX Settings Architecture & Technical Specification Documentation prior reading this document.

Naming Conventions

Lowercase is a naming convention in which a name formed of a single word is written all letters in lowercase.
Example: name, version, uuid, etc.

Uppercase is a naming convention in which a name formed of a single word is written all letters in uppercase.
Example: HOME, SHELL, PATH, etc.

Titlecase is a naming convention in which a name is written with all letters in lowercase except its first letter, which is uppercase. It follows a more natural style. No blank space allowed.
Example: Chars, Dimension, Geometry, etc.

Camelcase is a naming convention in which a **name** is formed of multiple words that are joined together as a single word with the first letter of each of the multiple words capitalized so that each word that makes up the **name** can easily be read. No blank space allowed.
Example: maximumSize, backgroundColor, darkColor, etc.

MaXX Settings Convention

The table below lists the naming convention used in MaXX Settings.

	Convention	Samples
Attribute	One or multiple words in **camelcase** without blank space..	version, maxDuration, defaultAppName
Stereotype	One word in the **titlecase** without blank space.	Chars, Geometry, Image
Schema Name	Multiple words with no blank space where each word is in the **titlecase**. The last word usually defines the Schema's Stereotype name.	TextColor, DoubleClickGauge, AccelerationGauge
Schema Filename	Multiple words with no blank space where each word is in the **titlecase**. The last word usually defines the Schema's Stereotype name and is separated with a period.	Username.Chars, DoubleClick.Gauge, Acceleration.Gauge

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

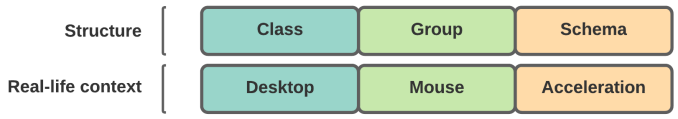
Instruments

In this section, we go over the nuts and bolts regarding Instruments, their uses as System wide and User Preference settings, and the way they are persisted by MaXX Settings. An Instrument is in simple terms a Schema file used in a context (or use-case) defined by Class/Group Combo classification. Schema is the abstraction of a set of values grouped together described by a particular concept of information, a Stereotype.

Refer to the MaXX Settings Architecture & Technical Specifications for more details.

Classification

One of the main design goals of MaXX Settings is to retrieve information as fast as possible and without introducing too much complexity in the process. So for this important performance requirement alone, MaXX Settings must provide an efficient mechanism for classifying and retrieving information. It is known that Instruments are made of a *Class.Group.Schema* structure could be mapped directly onto the file system with physical directories and files. Instead, MaXX Settings use an ultra fast computable hashcode of the Instrument's name, then mapped into a hashed directory structure. This allows lightning fast lookup regardless of the number of stored elements and is less prone to manual human intervention (messing things up). This is the way...



Instrument Structure vs. real-life

Instrument Categories

Category	Scope	Class Name	Group Name
User Experience	**System Wide Instruments** **User Preference Instruments**	**Desktop**	**Mouse** **KeyboardSettings** **KeyboardShortcuts** **Background** **DtUtilities** **Window** **Settings** **Colors** **DtSounds** **Localization** **Text** **FontRendering** **FileManager** **IconCatalog**
FileTypes		**FileTypes**	
Applications		**Application**	**WinEditor** **ImageEditor** **ImageViewer** **...**

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

System Wide Instruments

As we saw previously, MaXX Settings Root directory is defined by the Environmental Variable **$MAXX_SETTINGS**. Therefore all MaXX Settings Instruments are stored in the **$MAXX_SETTINGS/Instruments** directory. Those **Instruments** are called **System Wide Instruments**, they are read-only for normal users and only modifiable via the Administrative Command Line Interface with superuser privilege.

System Wide Nomenclature

Let's explore the System wide Instrument Desktop.Mouse.Acceleration and its various properties.	
Name	Desktop.Mouse.Acceleration
Class	Desktop
Group	Mouse
Schema	Acceleration
Stereotype	Gauge
Schema File	Acceleration.Gauge
Fully Qualified Name (FQ Name)	/Desktop/Mouse/Acceleration.Gauge
Hashed Storage Location	/3b/2a/d4/d6
Physical File Path	$MAXX_SETTINGS/Instruments/3b/2a/d4/d6/Acceleration.Gauge

System Wide Instruments Location

$Root: $MAXX_SETTING/**Instruments**
$Filename: $Root/$Classification/***<Schema>.<Stereotype>***

Example

$Filename: /opt/MaXX/share/msettings/**Instruments**/14/ab/58/**Acceleration.Gauge**

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

User Preference

We know already System Wide Instruments are read-only from a normal user point of view since they only define validation rules and default values. So how do we handle custom preferences for one or multiple users on the same system? The solution is rather simple, we just don't use them for say, but rather extend them and reusing the same classification strategy **<Class>.<Group>.<Schema>** for storing only the user defined values, but in a user specific location. Basically, they are user-land **Instruments** that can be editable by normal users, a.k.a. **User Preferences**.

By default User Preferences are located inside the **$HOME/.maxxdesktop/msettings/Preferences** directory and follow the same storage convention as System wide Instruments.

User Preferences are sharing the same classification and hashed storage location structure as System wide Instruments. This also means that the calculated hashcodes are the same.

User Preference Nomenclature

Let's explore an User Preference Instrument Desktop.Mouse.Acceleration and its various properties.	
Name	Desktop.Mouse.Acceleration
Class	Desktop
Group	Mouse
Schema	Acceleration
Stereotype	Gauge
Schema File	Acceleration.Gauge
Fully Qualified Name (FQ Name)	/Desktop/Mouse/Acceleration.Gauge
Hashed Storage Location	/3b/2a/d4/d6
Physical File Path	$HOME/.maxxdesktop/msettings/Preferences/3b/2a/d4/d6/Acceleration.Gauge

User Preference Instruments Location

$URoot: $HOME/.maxxdesktop/msettings/**Preferences**/
$Filename: $URoot/$Classification/***Schema.Stereotype***

Example

$Filename: $HOME/.maxxdesktop/msettings/**Preferences**/14/ab/58/**Acceleration.Gauge**

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Desktop Instrumentation

<need some text here>

User Experience Desktop Choices

Desktop Choices are a set of predefined Complex Choices that supports User Experience Instruments. Mosty from the Desktop classification.

Desktop Choices

Schema	Option Type	Options	Description / Comment
ColorSpace.Choice	Chars	RGB255, RGB100, YUV, HSL, CMYK,	Refer to Apple Developer Site
FontStyle.Choice	Chars	Normal	
FontWeight.Choice	Chars	Light, Medium, Demibold, Bold, Black	
FontSlant.Choice	Chars	Italic, Oblique, Roman	
Language.Choice	Chars	...	List of supported Language
KeyboardInput.Choice	Chars	...	List of supported KeyboardInput
DefaultSoundOutput.Choice	Chars	...	List of supported Sound Output Devices ??
SGIScheme.Choice	Chars	Arizona, Bayou,BlackAndWhite, DarkBliss, Gainsborough, Gotham, IndigoMagic, Inverness, Lascaux, Leonardo, Metropolis, Milan, Pacific, Potrero, RedGreenSafe, Rembrandt, RoseGarden, Sargent, VanGogh, Willis, Buckingham, GrayScale, KeyWest, Mendocino, Monet, Print, Rio, Titian, Turner, Vancouver, Whistler	List of supported SGI Schemes
IconSortBy.Choice	Chars	Name, Type, Size, CreationDate, ModifiedDate	
IconViewAs.Choice	Chars	Icon, List, Detail	
XftLcdFilter.Choice	Chars	lcddefault, lcdlight, lcdnone, lcdlegacy	The ***lcddefault*** filter will work for most users. Other filters are available that can be used in special situations: ***lcdlight***; a lighter filter ideal for fonts that look too bold or fuzzy, ***lcdlegacy***, the original Cairo filter; and ***lcdnone*** to disable it entirely.
XftHintStyle.Choice	Chars	hintnone, hintslight, hintmedium, hintfull	While **hintfull** will be a crisp font that aligns well to the pixel grid but will lose a greater amount of font shape. **hintslight** implicitly uses the **autohinter** in a vertical-only mode in favor of font-native information for non-CFF (*.otf*) fonts.
XftRgba.Choice	Chars	rgb, bgr, vrgb, vbrg	Monitors are either: **RGB** (most common), **BGR**, **V-RGB** (vertical), or **V-BGR**. A monitor test can

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

			be found here.
WinEditor.Choice	Command	NEdit, XNEdit, Gedit	
FileBrowser.Choice	Command	Rox-filer, fm	
ImageViewer.Choice	Command	Feh, Eog	
ImageEditor.Choice	Command	gimp	
WebBrowser.Choice	Command	firefox, chrome	
EmailClient.Choice	Command	thunderbird, evolution	
MediaViewer.Choice	Command	vlc	
VectorEditor.Choice	Command	inkscape	
PDFViewer.Choice	Command	Xpdf, evince	
BackgroundColors.Choice	Image	...	List of predefined Background Images
BackgroundImages.Choice	Color	...	List of predefined Background Colors

Note: 24 total User Experience Choices

Desktop Choices Location

$Root: $MAXX_SETTING/**Choices**
$Filename: $Root/***<Schema>.Choice***

Example - WinEditorChoice

$Filename: /opt/MaXX/share/msettings/**Choices**/WinEditor.Choice

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Applications Instrument

FileTypes are using the same classification and lookup mechanism as Desktop User Experience Instruments.

Application

Class	Group	Schema	Name
Application	**WinEditor**	NEdit.Command	NEdit
		Gedit.Command	Gedit
		XNEdit.Command	XNEdit
	ImageEditor	GIMP.Command	GIMP
	ImageViewer	ImageViewer.Command	ImageViewer
		Feh.Command	Feh
		EOG.Command	EOG

WORK IN PROGRESS :)

FileType User Experience Instruments

MaXX Settings provides an extensible mechanism to associate file types and actions with corresponding applications in a truly limitless way. FileType Instruments offers a powerful file type to application matching engines based on MIME types, file extensions and content matching rules with programmable actions like view, edit, run, print, compile, etc.

FileTypes are using the same classification and lookup mechanism as Desktop User Experience Instruments.

Classification : FileType
Group: (SUPERTYPE)
Schema : (Type)

Instruments Location

$Root: $MAXX_SETTING/**Instruments**
$Classification: **/<**Class>/<Group>/<Schema>.<Stereotype> ***which is a calculated Hashed Directory Structure***
$Filename: **$Root/$Classification/**<Schema>.<Stereotype>

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

FileTypes

Class	Group	Schema	Description
FileType	**SUPERTYPE**	*Type.Application*	Command to execute to View, Edit or Run a specific Class/Group filetype.
	Text	Plain.Application XML.Application JSON.Application	
	Audio	x.Application Edit.Application	
	Video	View.Application Edit.Application Run.Application	

SUPERTYPE The type-name is the TYPE name of any valid file type. Use SUPERTYPE to identify the file type as a "subset" of one or more other file types. This information can be accessed by other file types by calling isSuper(1) from within their CMD rules (OPEN, ALTOPEN, and so on). A file type can have multiple SUPERTYPEs. (For example, the Script file type has both Ascii and SourceFile SUPERTYPES.)

Example: **SUPERTYPE** *Executable*

TYPE The type-name is a one-word ASCII string. You can use a legal C language variable as a type name. Choose a name that is in some way descriptive of the file type it represents. All rules that follow a **TYPE** declaration apply to that type, until the next **TYPE** declaration is encountered in the FTR file. Each Type declaration must have a unique Type name.

Example: **TYPE** *GenericExecutable.*

Text/Plain.Application

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Implementation Details

This section focuses on the implementation details of MaXX Desktop Preference Panels, how they integrate with MaXX Settings and finally how they apply those settings.

MaXX Settings Integration

<need some description>

<how the integration is done>
<C++ class diagram>

<how settings are applied>
<C++ class diagram>

Desktop Class Instruments

<need some description>

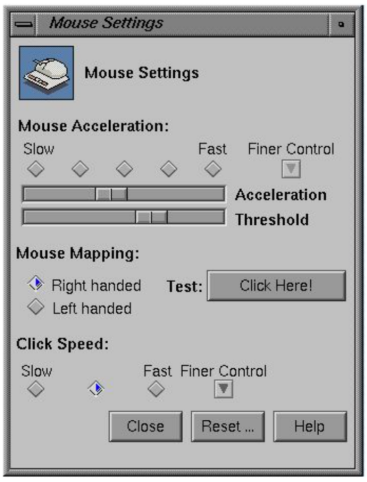
Breakdown Summary
13 Groups
92 total User Experience Instruments

WIP

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Mouse Settings

Classification: Desktop
Group: Mouse

Schema	Default	Preference Panel
Acceleration.Gauge	minimum=0.0 maximum=10.0 scale=1.0 default=2.0	
Threshold.Gauge	minimum=0.0 maximum=15.0 scale=1.0 default=4.0	
LeftHanded.Logical	false	
WheelMouseScroll.Logical	true	
NaturalScrolling.Logical	false	
DoubleClick.Gauge	minimum=0.0 maximum=10.0 scale=1.0 default=4.0	

Links:
https://wiki.archlinux.org/index.php/Mouse_acceleration
https://askubuntu.com/questions/172972/configure-mouse-speed-not-pointer-acceleration
https://tronche.com/gui/x/xlib/input/keyboard-and-pointer-settings.html
https://tronche.com/gui/x/xlib/input/XChangePointerControl.html
/home/emasson/MaXX-Dev/cdesktopenv-code/cde/programs/dtstyle

XLib function calls:

- XChangePointerControl
- XGetPointerControl
- XtGetMultiClickTime
- XtSetMultiClickTime

Todo:

- Define Gauge values (table above) and feature+validate good default values. DoubleClick might be using decimals
- Write CLI commands (script)
- Integrate msettings CLI into **mouse** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **mouse** application (does not start the visual portion of the app)
- Add logic to apply settings with XLib calls above

```
$ xset q | grep -A 1 Pointer
  acceleration:  2/1  threshold:  4
```

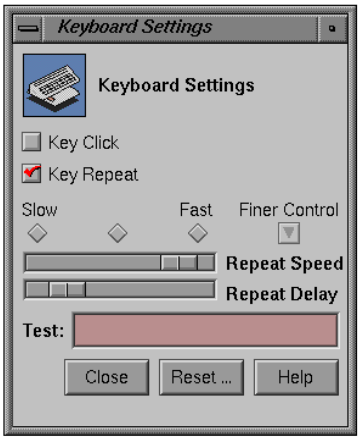
The **acceleration** is a *multiplier number* that defines how many times faster than the standard speed the cursor will move. Try numbers between 2 and 5, setting a high multiplier like 9 makes the mouse movements very jumpy. It does not need to be a whole number, you can use 1/2 to get half the standard speed or 5/2 (=2.5) if 2 is too slow and 3 is too fast.

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

The **threshold** defines how many pixels the mouse must move in a short period of time before the acceleration setting is used. Using a threshold of 1, as in xset m 5 1, disables this and gives you the same mouse speed at all the time. Setting xset m 5 10 requires the mouse to move 10 pixels before the pointer is accelerated.

Keyboard Settings

Classification: Desktop
Group: KeyboardSettings

Schema	Default	Preference Panel
KeyClick.Logical	true	
KeyRepeat.Logical	true	
RepeatSpeed.Gauge	-	
RepeatDelay.Gauge	-	
KeyClickVolume.Gauge	-	

Links:
https://wiki.archlinux.org/index.php/Xorg/Keyboard_configuration
https://tronche.com/gui/x/xlib/input/keyboard-and-pointer-settings.html

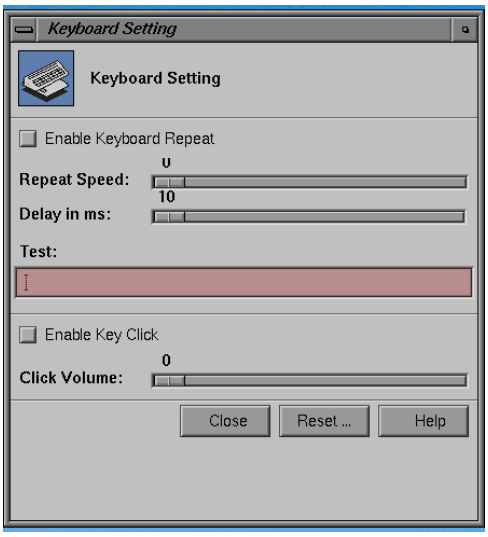
XLib function calls:

- XSetInputFocus()
- XGetInputFocus()
- XChangeKeyboardControl()
- XGetKeyboardControl()
- XAutoRepeatOn
- XAutoRepeatOff

Todo:

- Define Gauge values (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **keyboard** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **keyboard** application (does not start the visual portion of the app)
- Add logic to apply settings with XLib calls above

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.



MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Keyboard Shortcuts

Classification: Desktop
Group: KeyboardShortcuts

Schema	Default	Preference Panel
KeyClick.Logical	true	NEW IMAGE
KeyRepeat.Logical	true	
RepeatSpeed.Gauge	-	
RepeatDelay.Gauge	-	

Links:
https://wiki.archlinux.org/index.php/Xorg/Keyboard_configuration
https://tronche.com/gui/x/xlib/input/keyboard-and-pointer-settings.html

XLib function calls:

- XSetInputFocus()
- XGetInputFocus()
- XChangeKeyboardControl()
- XGetKeyboardControl()
- XAutoRepeatOn
- XAutoRepeatOff

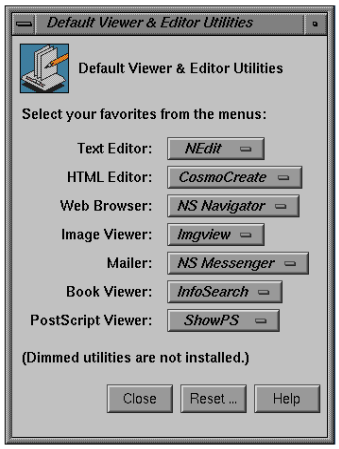
Todo:

- Define Gauge values (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **keyboard** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **keyboard** application (does not start the visual portion of the app)
- Add logic to apply settings with XLib calls above

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Default Application Settings

Classification: Desktop
Group: DtUtilities

Schema	Default	Preference Panel
FileBrowser.Choice	-	
WinEditor.Choice	-	
TextEditor.Choice	-	
WebBrowser.Choice	-	
EmailClient.Choice	-	
ImageEditor.Choice	-	
ImageViewer.Choice	-	
MediaViewer.Choice	-	
VectorEditor.Choice	-	
PDFViewer.Choice	-	

Links:

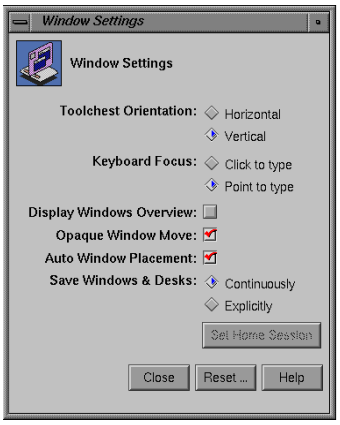
Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **dtutilities** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **dtutilities** application (does not start the visual portion of the app)
- Edit **$HOME/.maxxdesktop/desktopenv** script to load settings via CLI command (to set export values)
- Add logic to apply settings then run **update-desktop** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Window Settings

Classification: Desktop
Group: Window

Instrument	Default	Preference Panel
ToolchestHorizontal.Logical	false	
KeyboardFocus.Logical	true	
DisplayOverview.Logical	false	
MoveOpaqueWindow.Logical	true	
OutlineThickness.Gauge	2 1-4:2	
WindowManagerAccent.Color	red	
AutoWindowPlacement.Logical	true	
SaveWindowsDesks.Logical	true	
Modern.WindowTitle.Typeface	-	
Modern.IconTitle.Typeface	-	
Classic.WindowTitle.Typeface	-	
Classic.IconTitle.Typeface	-	
MoveWithoutRaising.Logical	true	

Links:

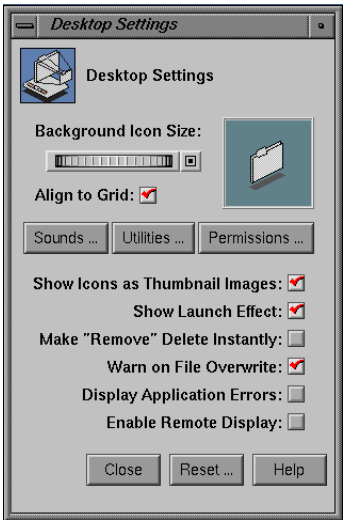
Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **window** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **window** application (does not start the visual portion of the app)
- Edit **$HOME/.maxxdesktop/desktopenv** script to load settings via CLI command (to set export values). Maybe extract some of the Xresources attributes (5Dwm and others).:
- Add logic to apply settings then run **update-desktop and tellwm** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Desktop Settings

Classification: Desktop
Group: Settings

Schema	Default	Preference Panel
DesktopIconSize.Gauge	-	
DesktopIconAlignGrid.Logical	true	
DesktopAccemt.Color	yellow	
ToolchestSoundEffect.Logical	true	
ModernLookAndFeel.Logical	false	
IconAsThumbnailImage.Logical	true	
ShowLaunchEffect.Logical	true	
MakeDeleteInstantly.Logical	false	
WarnOnFileOverwrite.Logical	true	
DisplayApplicationErrors.Logical	false	
EnableRemoteDisplay.Logical	false	

Links:

Todo:

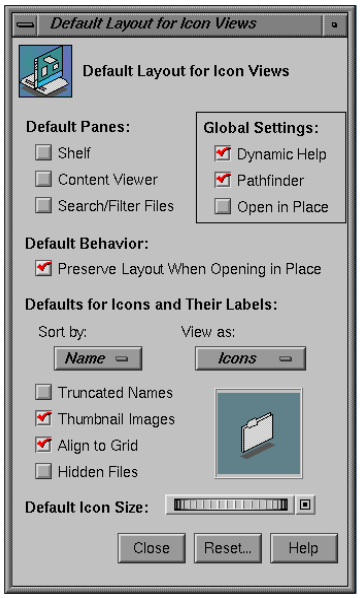
- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **workspace** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **workspace** application (does not start the visual portion of the app)
- Add logic to apply settings then run **update-desktop and tellworkspace** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

FileManager Settings

Classification: Desktop
Group: FileManager

DisplayShelf.Logical	false
DisplayContent.Logical	false
DisplaySearchFilters.Logical	false
KeepLayoutOpenInPlace.Logical	true
IconSortBy.Choice	-
IconViewAs.Choice	-
TruncateNames.Logical	false
ThumbnailImages.Logical	true
AlignToGrid.Logical	true
DisplayHiddenFiles.Logical	false
DefaultIconSize.Gauge	-
IconText.Typeface	-
DynamicHelp.Logical	trre
PathFinder.Logical	true
OpenInPlace.Logical	false



Links:

Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **fm** application
- Add logic to apply settings then run **update-desktop** and restart **fm** afterwards.

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

IconCatalog Settings

Classification: Desktop
Group: IconCatalog

KeepLayoutOpenInPlace.Logical	false	NEW IMAGE
IconSortBy.Choice	-	
IconViewAs.Choice	-	
TruncateNames.Logical	false	
ThumbnailImages.Logical	true	
AlignToGrid.Logical	true	
DefaultIconSize.Gauge	-	
IconText.Typeface	-	

Links:

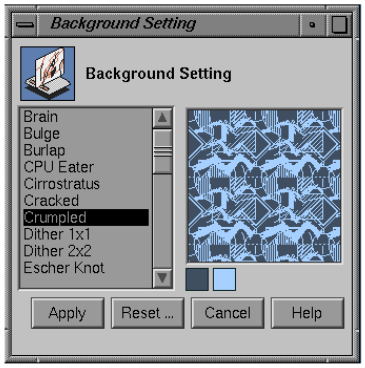
Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **iconcatalog** application
- Add logic to apply settings then run **update-desktop** and restart **iconcatalog** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Background Settings

Classification: Desktop
Group: Background

Schema	Default	Preference Panel
BackgroundColors.Choice	-	
BackgroundImages.Choice	-	
DarkBackground.Logical	true	
Pattern1.Color	-	
Pattern2.Color	-	
Pattern3.Color	-	

Links:

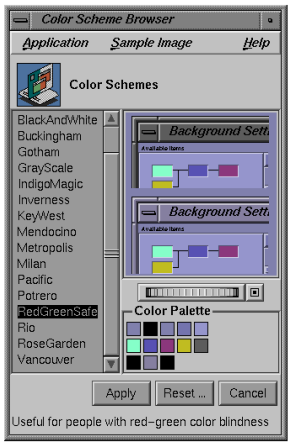
Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **background** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **background** application (does not start the visual portion of the app)
- Add logic to apply settings then run **update-desktop** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Scheme Settings

Classification: Desktop
Group: Scheme

Schema	Default	Preference Panel
SGIScheme.Choice	IndigoMagic	
SgiDarkScheme.Logical	false	
UserInterfaceAccent.Color	blue	

Links:

Todo:

- Define the Color (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **scheme** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **scheme** application (does not start the visual portion of the app)
- Maybe check current $HOME/.maxxdesktop/desktopenv and Xresource script
- Add logic to apply settings then run **update-desktop and tellwm** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Text Settings

Schema	Default	Preference Panel
SmallText.Typeface	-	NEW IMAGE
NormalText.Typeface	-	
LargeText.Typeface	-	
Terminal.Typeface	-	
WindowTitle.Typeface	-	
WindowIconTitle.Typeface	-	
SmoothText.Logical	True	
Links:		

Todo:

- Define the Typeface (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **???** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **???** application (does not start the visual portion of the app)
- Add logic to apply settings then run **update-desktop** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Font Rendering Settings

Schema	Default	Preference Panel
XftAutoHint.Logical	0	NEW IMAGE
XftLcdFilter.Choice	lcddefault	
XftHintStyle.Choice	hintslight	
XftHinting.Logical	1	
XftAntialias.Logical	1	
XftRgba.Choice	rgb	

Links:

Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **???** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **???** application (does not start the visual portion of the app)
- Add logic to apply settings then run **update-desktop and tellwm** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Sound Settings

Schema	Default	Preference Panel
MuteSystem.Logical	false	
StartupShutdownTunes.Logical	true	
DesktopSounds.Logical	true	
SystemAlertsSounds.Logical	true	
KeyboardBell.Logical	true	
KeyClickVolume.Gauge	-	
DefaultSoundOutput.Choice	false	
SoundOutputVolume.Gauge	-	

Links:

- http://blog.chapagain.com.np/ubuntu-linux-increase-decrease-volume-from-command-line-keyboard-shortcut/

Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **dtsounds** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **dtsounds** application (does not start the visual portion of the app)
- Add logic to apply settings then run **update-desktop** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Localisation Settings

Schema	Default	Preference Panel
Language.Choice	-	
KeyboardInput.Choice	-	
Links:		

Todo:

- Define the Choices (table above)
- Write CLI commands (script)
- Integrate msettings CLI into **??** application
- Add command-line option **-apply** that just start, load, apply the settings and quit the **??** application (does not start the visual portion of the app)
- Add logic to apply settings then run **update-desktop and tellwm** afterwards

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

External References

Colors Settings
https://en.wikipedia.org/wiki/Color_model
https://en.wikipedia.org/wiki/Color_space

https://stackabuse.com/reading-and-writing-yaml-files-in-java-with-jackson/

https://overiq.com/c-programming-101/array-of-strings-in-c/

https://wiki.archlinux.org/index.php/Font_configuration

https://feh.finalrewind.org/

https://imagemagick.org/index.php

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

Code Snippets

Read one time	Read text file line by line C++

`std::string buffer;`
 	
`std::ifstream f("file.txt");`
`f.seekg(0, std::ios::end);`
`buffer.resize(f.tellg());`
`f.seekg(0);`
`f.read(buffer.data(), buffer.size())`
 	
`#include <dirent.h>`
 	
`if (auto dir = opendir("some_dir/")) {`
`  while (auto f = readdir(dir)) {`
`  if (!f->d_name || f->d_name[0] == '.')`
`  continue; // Skip everything that starts with a dot`
 	
`  printf("File: %s\n", f->d_name);`
`  }`
`  closedir(dir);`
`}`
	`bool loadMenu(const char* filename) {`
	`ifstream ifs(filename, ios::in);`
	`  if (ifs.fail()) {`
	`  cout << "Toolchest: Menu file " << filename << " not found" << endl;`
	`  return false;`
	`  }`
	
	`  while (!ifs.eof()) {`
	`  string buffer;`
	`  getline(ifs, buffer, '\n');`
	`  //ignore blank lines`
	`  if (buffer.size() == 0) continue;`
	
	`  string *data = new string(buffer);`
	`  if ((data->find("#") == 0) || (data->find("!") == 0)) {`
	`  // comment line, do nothing`
	`  } else {`
	`  //add the line to the array!`
	`  theFile[elements] = data;`
	`  elements++;`
	`  }`
	`  }`
	`  ifs.close();`
	`  return true;`
	`}`

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.

HashCode FilePath Generator in C++	HashCode FilePath Generator in Java

`#include <iostream>`
`#include <string>`
`#include <ctime>`
`#include <chrono>`
 	
`int hashCode(std::string);`
`int main()`
`{`
 	
`  auto start = std::chrono::steady_clock::now();`
 	
`  std::string str = "Desktop.Settings.DisplayApplicationErrors";`
`  std::size_t hash = hashCode(str);`
 	
`  int mask = 255;`
`  int firstDir = hash & mask;`
`  int secondDir = (hash >> 8) & mask;`
`  int thirdDir = (hash >> 8 >> 8) & mask;`
`  int forthDir = (hash >> 8 >> 8 >> 8) & mask;`
 	
`  auto end = std::chrono::steady_clock::now();`
`  auto elapsed = std::chrono::duration_cast<std::chrono::nanoseconds>(end - start);`
`  std::cout << "It took me " << elapsed.count() << " nanoseconds." << std::endl;`
 	
`  char filepath[50];std::sprintf( filepath, "%02x/%02x/%02x/%02x", firstDir, secondDir, thirdDir, forthDir );`
`  std::cout << str << " hash=" << hash <<  " path=" <<  filepath << std::endl;`
`}`
 	
`g++ -std=c++11 hash.cpp -o hash -L/usr/lib64 -lstdc++ -lm`
	`String str = "Desktop.Settings.DisplayApplicationErrors";`
	
	`int hash = hashCode();`
	
	`  int mask = 255;`
	`  int firstDir = hash & mask;`
	`  int secondDir = (hash >> 8) & mask;`
	`  int thirdDir = (hash >> 8 >> 8) & mask;`
	`  int forthDir = (hash >> 8 >> 8 >> 8) & mask;`
	
	`String path = String.format("%02x/%02x/%02x/%02x", firstDir, secondDir, thirdDir, forthDir);`

`public int hashCode(std::string value) {`
`  int h = 0;`
`  int s = value.size();`
`  char c_str[s + 1];`
`  strcpy(c_str, value.c_str());`
`  if (h == 0 && s > 0) {`
`  for (int i = 0; i < s; i++) {`
`  h = 31 * h + c_str[i];`
`  }`
`  }`
`  return h;`
`}`
	`public static int hashCode(String strValue) {`
	`  int h = 0;`
	`  char[] value = strValue.toCharArray();`
	`  if (h == 0 && value.length > 0) {`
	`  for (int i = 0; i < value.length; i++) {`
	`  h = 31 * h + value[i];`
	`  }`
	`  }`
	`  return h;`
	`}`

`Average time 4500 nanoseconds on Xeon 5690 @4.13Ghz`	`Average time 143324 nanoseconds on Xeon 5690 @4.13Ghz` `Average time 6700 nanoseconds on Xeon 5690 @4.13Ghz (native)`

MaXX Interactive Desktop © 2020, 2021 - All rights reserved.