FAQ

Here is a list of the frequently asked questions:

---

1. General

1.1 How do I run a program on startup/autostart?

Let's say you want "urxvt":http://software.schmorp.de/pkg/rxvt-unicode.html to start after [[subtle]], and for some reason @echo "urxvt" >> ~/.xinitrc@ is just not cutting it. Using the *:start* hook and [[subtlext]] we can simulate autostart like so:

<pre>{{hide}}<code class="ruby">
on :start do
  Subtlext::Subtle.spawn "urxvt"
end
</pre></code>

1.2 How do I set a wallpaper in subtle?

On start, [[subtle]] sets a background color (_if set_) to the root window and therefore *overwrites* any root pixmap set before. To avoid this, you just need to comment out the *:background* line from your config.

[[subtle]] itself has no and will never have either an autostart or way to set a wallpaper directly. Normally you just want to setup your X session and and not [[subtle]]. Your _~/.xinitrc_ is the right place for stuff like this.

A background can easily set with a tool like "feh":http://linuxbrit.co.uk/software/feh/.

1.3 Is subtle a reparenting window manager?

Nope, [[subtle]] doesn't reparent windows and there is in fact *no* reason to do that. The layout in [[subtle]] is a really loose set, the only relation between a [[views|view]] and a [[client]] is [[tagging]] and this is checked on every [[views|view]] update.

Reparenting windows would require following additional steps, when a window is visible on a certain [[views|view]]:

Resize the [[views|view]] toplevel window to the size of the current [[screen]]
Reparent the [[client]] window to the toplevel window
Handle (ignore here) the generated expose and crossing events

Probably sounds like not much overhead, but keep in mind this is just required because the developer of "Java":http://www.java.com/ and "Qt":http://qt.nokia.com/products/ cannot understand following line from the "ICCCM":http://tronche.com/gui/x/icccm/sec-4.html#s-4.2.1:

bq. Clients must be aware that some window managers will reparent their top-level windows so that a window that was created as a child of the root will be displayed as a child of some window belonging to the window manager

1.4 Is there any log file?

Nope, there is *no* log file. In case you need one for e.g. reporting a bug +please+ read to the [[bugs|reporting a bug]] wiki page and check the paragraph about [[bugs#enable-logging|logging]].

1.5 Why does subtle cause to many wakeups?

The wakeups are caused by a "Ruby":http://ruby-lang.org internal polling thread that runs every 10ms to schedule/handle other threads and signals. There is currently no way to avoid that in "YARV":http://www.atdot.net/yarv/, although the problem is well known.

http://www.ruby-forum.com/topic/200147

There is finally some progress regarding this: http://www.ruby-forum.com/topic/549998

2. Keys

2.1 How can I use subtle without numpad?

Per default [[subtle]] uses the numpad, because it's easier to remind the different postions when you can see them. Generally this is just a suggestion, it's up to you to change the grabs to your needs - any keys will work.

Keys that also have proven to work well are *q-w-e*, *a-s-d* and *y-x-c*:

<pre>{{hide}}<code class="ruby">grab "W-q", [:top_left,   :top_left66,   :top_left33   ]
grab "W-w", [:top,   :top66,   :top33   ]
grab "W-e", [:top_right,   :top_right66,   :top_right33   ]
grab "W-a", [:left,   :left66,   :left33   ]
grab "W-s", [:center,   :center66,   :center33   ]
grab "W-d", [:right,   :right66,   :right33   ]

QWERTZ
grab "W-y", [:bottom_left, :bottom_left66, :bottom_left33]

QWERTY
grab "W-z", [:bottom_left, :bottom_left66, :bottom_left33]

grab "W-x", [:bottom,   :bottom66,   :bottom33   ]
grab "W-c", [:bottom_right, :bottom_right66, :bottom_right33]
</code></pre>

You can find more about assigning keys [[Grabs|here]].

2.2 What is the difference between ViewSwitch and ViewJump?

Boths keys change the [[view]] of the current active [[screens|screen]], but the behavior how they do that depends on if the system consists either of a single [[screens|screen]] or more.

h2. Single screen

Here both [[grabs]] do +exactly+ the same.

h2. Multi screen

Here the behavior depends on if the selected [[views|view]] is visible on another [[screens|screen]] or not.

* *ViewSwitch* either swaps the current [[views|view]] with the selected if it is visible on another [[screens|screen]] or just sets the selected [[views|view]] on current [[screens|screen]].
* *ViewJump* either focus the selected [[views|view]] if it is visible on another [[screens|screen]] or just sets the select [[views|view]] on current [[screens|screen]].

2.3 Why don't my grabs work with xyz keyboard layout?

Actually, I have no idea why setting the keyboard layout in your @xorg.conf@ isn't suffucient, but apparently you need to call *setxkbmap* in one of your startup files to set the keymap. My suggestion is to use the @~/.xinitrc@, because it is perfectly suited to setup Xorg and usually read.

Just add something like this to your @~/.xinitrc@ *before* you start [[subtle]]:

<pre><code>setxkbmap -layout 'de(nodeadkeys)'</code></pre>

3. Programs

3.1 How do I (auto)start programs?

There are several way how to start a certain programm, here are the most common:

* Start your program via your $HOME/.xinitrc:

<pre><code class="ruby">
subtle &
PID=$!
sleep 2
urxvt
wait $PID
</code></pre>

* Start your program via [[grabs]]:

<pre><code class="ruby">
"A-x" => "urxvt"
</code></pre>

There are many launchers available like "dmenu":http://tools.suckless.org/dmenu and there is even a launcher especially designed for [[subtle]], you can find it in [[subtle-contrib:Launcher|here]].

3.2 How do I run a Java program?

"Java":http://java.com *expects* a certain behaviour (_reparenting of client windows_) of a window manager which is not part of any standard, therefore some "Java":http://java.com programs just show a white canvas.

If this happens just try to start your program like this:

<pre><code class="shell">AWT_TOOLKIT=MToolkit program</code></pre>

This changes the default tookit of "Java":http://java.com to _MToolkit_, which is known to work with non-reparenting windows managers like [[subtle]]. Depending on your "OpenJDK":http://openjdk.java.net version and your architecture this may either lead to a segmentation fault or your "OpenJDK":http://openjdk.java.net just has no support for *MToolkit*. In this case check if your distribution has applied a patch for this issue which allows something like this to change the default behaviour:

<pre><code class="shell">_JAVA_AWT_WM_NONREPARENTING=1 program</code></pre>

In case both doesn't work there is a third option: "Java":http://java.com seems to use an internally hardcoded list of window managers that are non-reparenting, the "suckless":http://suckless.org guys made a tool ("wmname":http://tools.suckless.org/wmname) to change the name of the wm.

Since r2520 [[subtle]] can do that for you, just enable the *:wmname* option.

Generally this problem is really long lasting, see here:

http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6511454
http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=508650

bq. Note: Many problems only affect the "JRE":http://www.oracle.com/technetwork/java/javase/downloads/index.html and can be avoided by using the "OpenJDK":http://openjdk.java.net.

3.3 Why does program XYZ appears/turns as/to a gray canvas?
Generally, this happens, when a program needs a specific aspect ratio and [[subtle]] sets a gravity that violates this. Windows can tell the window manager about this kind of preferences via their size hints and [[subtle]] can honor those.

There are two possible ways:

Enable size hints globally with the _:resize_ option
<pre>{{hide}}<code class="ruby">set :resize, true</code></pre>
Enable this per program via [[tagging|tag]]
<pre>{{hide}}<code class="ruby">tag "name" do
match "name"
resize true
end</code></pre>

bq. Note: This can happen with "Java":http://java.com too, see "here":http://subforge.org/ezfaq/show/subtle?faq_id=14 for more information.

4. Tagging
4.1 How do I move a program to another view?
Placement in [[subtle]] is +strict+ and completely done via [[tagging]]. There are many ways to change [[tagging|tags]] per runtime, common is to use either [[subtler]] or [[subtlext]].

h2. subtler

[[subtler]] can be used on the commandline:

<pre>{{hide}}<code class="ruby">
subtler -ta tag #< Add new tag 'tag'
subtler -cT client tag #< Tag client 'client' with tag 'tag'
subtler -vT view tag #< Tag view 'view' with tag 'tag'
</code></pre>

h2. subtlext

[[subtlext]] requires basic "ruby":http://ruby-lang.org knowledge:

<pre>{{hide}}<code class="ruby">require "subtle/subtlext"

tag = Subtlext::Tag.new("tag").save #< Add new tag 'tag'
Subtlext::Client["client"] + "tag" #< Tag client 'client' with tag 'tag'
Subtlext::View["view"] + "tag" #< Tag view 'view' with tag 'tag'

</code></pre>

h2. Snippets

The [[snippets]] wiki page includes examples how to [[snippets#Move-windows|move to another view]].

h2. Contrib

h3. Vitag

"subtle-contrib":http://subforge.org/projects/subtle-contrib/wiki contains [[subtle-contrib:vitag|vitag]] a script to change the [[tagging|tags]] of windows and views with an editor.

h3. Launcher

The "launcher":http://subforge.org/projects/subtle-contrib/wiki#Launcher uses quite the opposite way, instead of moving a window to a certain screen it just provides a way to launch a window directly on the right view with the correct [[tagging|tags]].

h2. Stick

Most of the time, setting the window to *stick* does the trick too. Stick just displays the window on all views until the mode is disabled again. This can be done with [[grabs]] (default keys: *W-s*) or with [[subtler]]. (click on the window: @subtler -cXS@)

4.2 How do I tag console based programs?
When console based programs like "irssi":http://irssi.org are started with a terminal like @urxvt -e irssi@, they usually just change the *WM_NAME* of the terminal and per default, [[subtle]] uses both of the *WM_CLASS* values for tagging.

The problem about that is, that the [[tagging]] is normally done before the terminal really starts the desired app. To avoid that all better known terminals support the *-name* argument which changes the instance value (first string) of the *WM_CLASS*: @urxvt -name irssi -e irssi@

Another problem is that terminals retain their class value (second string) of the *WM_CLASS* and may match other [[tagging|tags]] like:

<pre><code class="ruby">tag "terms" do
  match "urxvt"
end</code></pre>

The common solution to bypass is this to use the :instance matcher, which just matches when for the specific instance value. So the easiest way to reliable match this client is a [[tagging|tag]] like this:

<pre><code class="ruby">tag "irssi" do
  match :instance => "irssi"
end</code></pre>

Following table shows the *WM_NAME* and *WM_CLASS* output of "xprop":http://www.xfree86.org/current/xprop.1.html of the various combinations:

_. Command	_. WM_NAME	_. WM_CLASS
urxvt	urxvt	urxvt, URxvt
urxvt -e irssi	irssi	urxvt, URxvt
urxvt -name irssi	urxvt	irssi, URxvt
urxvt -name irssi -e irssi	irssi	irssi, URxvt

Please have a look at the [[Tagging|tagging wiki page]] for more information.

4.3 How do I tag gimp?
[[Tagging]] "GIMP":http://gimp.org can be nasty, because the current version (2.6) uses inconsistent window names and roles. The window roles are fixed in the current development snapshot and will hopefully find their way into the 2.8 release.

https://bugzilla.gnome.org/show_bug.cgi?id=645456

4.4 How does subtle match clients?
[[subtle]] matches the {{color(#ff0000, WM_NAME)}} and the {{color(#0000ff, WM_CLASS)}} property of a window and not the title. This to ensure the match is correct.
There are several ways to get these values, here are the most common:

Select the window with "xprop":http://www.xfree86.org/current/xprop.1.htmland have a look for the {{color(#ff0000, WM_NAME)}} and {{color(#0000ff, WM_CLASS)}} lines which usually look like this:

<pre><code class="bash">{{color(#0000ff, WM_NAME)}}(STRING) = "urxvt"
{{color(#ff0000, WM_CLASS)}}(STRING) = {{color(#aa0000, "irssi")}}, {{color(#ff00ff, "URxvt")}}</code></pre>

Run *[[subtler]] -cl* and look for the matching line:

<pre><code class="bash">0x800007 * 1 1020x374 2 1 --- {{color(#ff0000, urxvt)}} ({{color(#0000ff,URxvt)}})</code></pre>

4.5 How to match a GLFW window
GLFW set the windows name after the window get created, so you can't match a particular GLFW window by his name.

to match any GLFW window :

<pre>
tag "glfw" do
 match name: "GLFW.*"
 # Your code....
end
</pre>

4.6 How to match a GLFW window
GLFW set the name of the window after the window get created, so you can't match a particular window by his name.

to match any GLFW window :

<pre>
tag "glfw" do
 match name: "GLFW.*"
 # your code
end
</pre>

4.7 Where is the output window of flash in fullscreen?
For flash, browsers seem to use window instance and class names that doesn't match the browser values. Therefore the windows will usually appear on the default view.

Following names are currently in use:

_. Browser	_. Arch	_. WM_NAME	_. WM_CLASS
Firefox <7.0.1	all	<unknown>	"<unknown>", "<unknown>"
Firefox >=7.0.1	all	plugin-container	"plugin-container", "Plugin-container"
Chromium	all	exe	"exe", "Exe"
Opera	x86	"operapluginwrapper"	"operapluginwrapper", "Operapluginwrapper"
nspluginwrapper	x86_64	"npviewer.bin"	"Npviewer.bin"

The easiest way to avoid that is to add a [[tagging|tag]] that makes these windows sticky:

<pre>{{hide}}<code class="ruby">tag "flash" do
  match "<unknown>|plugin-container|exe|operapluginwrapper|npviewer.bin"
  stick true
end</code></pre>

5. sur

5.1 How can I manually delete a sublet?
[[Sublets]] usually consist of a **.rb* file, a **.spec* file and one or more icons. All these files can be found in their respective folder in @$XDG_DATA_HOME/subtle@ resp. @~/.local/share/subtle@. After deleting the [[sublets|sublet]] files the cache of [[sur]] needs to be updated or otherwise [[sur]] will think the [[sublets|sublet]] is still installed: @sur update -l@

5.2 Where does sur store sublets and icons?
[[sur]] follows the "XDG Base Directory specifications":http://standards.freedesktop.org/basedir-spec/basedir-spec-latest.html to store data, which includes some base path for specific data types.

Following paths are used:

_. Type	_. XDG path	_. Default path
Sublets	@$XDG_DATA_HOME/subtle/sublets@	@~/.local/share/subtle/sublets@
Icons	@$XDG_DATA_HOME/subtle(icons@	@~/.local/share/subtle/icons@

6. Sublets

6.1 What is required for the volume sublet?
The volume [[sublets|sublet]] uses @/dev/mixer@ to set/get the volume levels, which is an older interface introduced with "OSS":http://www.opensound.com/ and still available via "ALSA OSS emulation":http://www.alsa-project.org/. Apparently, newer kernels refuse to autoload that module anymore and you need to load it manually or via any init file.

{{info(Please check to the docs of your distribution how to do it.)}}

<pre><code>modprobe snd_mixer_oss</code></pre>

{{warn(Following explanation is technical!)}}

The [[sublets|sublet]] needs a way to access a mixer without any asynchronous callbacks. The reason for that is that [[subtle]] is single-threaded and can't use a dedicated thread to wait for the reply, but the APIs of "ALSA":http://www.alsa-project.org/ and "PulseAudio":http://pulseaudio.org are both designed to be asynchronous. Normally event drivven is fine but it is troublesome when you can't use their mainloop.

Since there is no way to add e.g. a control socket to the event main loop of [[subtle]], the @/dev/mixer@ approach is the only way and works for all sound systems.

7. Not categorized

7.1 How to match a GLFW window

GLFW set the name of the window after the window get created, so you can't match a particular window by his name.

to match any GLFW window :

<pre>
tag "glfw" do
 match name: "GLFW.*"
 # your code
end
</pre>