

メニューや端末のウィンドウといった特定の種類のウィンドウを半透明にしたい。
システム監視プログラムやIMのコンタクトリストを、デフォルトですべてのデスクトップ上に表示するようにしたい。
ブラウザの新しいウィンドウを開くと、小さすぎたり大きすぎたりして困る。

そのような場合のために、特定のウィンドウが開かれたとき、半透明にしたり、スティッキーにしたり、一定サイズにしたりするルールをあらかじめセットしておくことが可能。
ウィンドウの照合条件を記述して自在に使いこなせるようにならなければ、Compiz Fusion を完全体にはできない。

注意：

このページの説明は、「[正規表現照合](#)」プラグインを有効にしておくことが前提。

ウィンドウの識別

Compiz には、ルール適用対象のウィンドウを識別する方法が 8 つある。
以下ではその 8 つの方法とともに、各方法に必要な情報を取得するための端末で実行するコマンドラインを列挙している。
コマンドを実行するとマウスカーソルが十字に変化するので、その十字を情報を取得したいウィンドウ上に置いてクリック。
なお、CCSM Ver.0.6.99 には xprop コマンドの GUI フロントエンド機能を持つルール編集ツールが組み込まれている。

・ タイプ (type) :

メニューやダイアログ、ヒント表示、その他といったウィンドウの種類。
種類を取得するためのコマンドは、`xprop|grep WINDOW_TYPE|cut -d_ -f10`

または次の中から選択：

unknown, combo, desktop, dialog, dnd, dock, dropdownmenu, fullscreen, modaldialog, menu, normal, notification, popupmenu, splash, toolbar, tooltip, utility

・ ロール (role) :

(限られたプログラムでのみ使用されているので、普通はブランク。)
ロールを取得するためのコマンドは、`xprop|grep "WM_WINDOW_ROLE(STRING)"|cut -d\" -f2`

・ ネーム (name) :

ウィンドウのクラス・ネーム。
ネームを取得するためのコマンドは、`xprop|grep WM_CLASS|cut -d\" -f2`

・ クラス (class) :

ウィンドウのクラス。
クラスを取得するためのコマンドは、`xprop|grep WM_CLASS|cut -d\" -f4`

・ タイトル (title) :

ウィンドウのタイトルバーに書かれている文字。
タイトルを取得するためのコマンドは、`xprop|grep "WM_NAME(STRING)"|cut -d\" -f2`

注意：

日本語のみの文字列をそのままルールとして入力しても動作しないようである。
しかし、日本語で始まるタイトルで、そのタイトルの頭から識別条件として入力する場合は、条件文字列の最後が半角アルファベット等で終わっているか、後述する正規表現が文字列の最後に用いられていれば機能する。
また、日本語の条件文字列の前後に正規表現を使用した場合も機能するようである。

・ XID (xid) :

ウィンドウのXID。すでに開いているウィンドウを識別するときには役に立たない。

・ ステイト (state) :

最大化や最前面表示、スティッキーといったウィンドウの状態。
次の中から選択：
modal, sticky, maxvert, maxhorz, shaded, skiptaskbar, skippager, hidden, fullscreen, above, below, demandsattention

`override_redirect` :

様々なプログラムによって様々な使用される属性、
これについては、この説明ページの最後の「特別なケース」の項を参照のこと。

ウィンドウは以下の形式の文字列で識別することが可能 :

上記のウィンドウの識別子=値

例えば、以下は「ポップアップ・メニュー」に該当する識別条件 :

`type=popupmenu`

注意 :

「 = 」の前後にはスペースを入れてはいけない

複数の識別条件を設定したい場合は、「 & 」が「かつ (And)」、「 | 」が「または (Or)」。
従って以下の例は「ドロップダウン・メニューまたはポップアップ・メニューのどちらかに該当する場合」を意味する :

`type=dropdownmenu | type=popupmenu`

「否定」条件を設定したい場合は、「 ! 」記号をウィンドウの識別子の頭に付ける。
以下の例は、「ダイアログ以外のすべてのウィンドウ」を意味する :

`!type=dialog`

識別条件には、違う種類の識別子を組み合わせ使用することもできる。

以下の例は、「ダイアログ・ウィンドウ以外で、ウィンドウのタイトルバーのタイトルが"konsole"という文字列で終わっているウィンドウ」に該当する :

`title=Konsole$ & !type=dialog`

"Konsole"の後ろに付いている「 \$ 」記号は正規表現であり、「 "konsole"で **終わっている** 文字列」を意味している。

例えば、「 ^Adobe Reader 」はその逆に「 "Adobe Reader"で **始まる** 文字列」を意味する。

" ^MPlayer\$ "のように前後をこれらの記号で囲むと、部分一致ではなく " MPlayer "の前後に文字列がない完全一致となる。

ちなみに正規表現を用いず単に " Konsole "とした場合、

上の例では「 " Konsole " をタイトルバーの文字列内のいずれかに含む、ダイアログ以外のすべてのウィンドウ」ということになる。

「 (」と「) 」を使用すると、複数の識別子を1つのセットとしてまとめることもできる。

以下の例は、「タイトルバーに " About " または " Configure"のどちらかの文字列が含まれているダイアログ・ウィンドウ」に該当する。

`type=dialog & (title=About | title=Configure)`

識別子を指定せず、単に値を入力をした場合は自動的に `type` の値として扱われる。

以下の例は両方とも同じ意味の識別条件となる :

`dropdownmenu | popupmenu | tooltip | unknown`
`type=dropdownmenu | type=popupmenu | type=tooltip | type=unknown`

最後に基本的な識別条件として挙げるのは「 any 」である。

「 any 」を指定するとすべてのあらゆるウィンドウが該当することになる。

逆に識別条件を一切記入せず空欄にしておくと、そのルールなどは機能しなくなる。

例えば、「 [General Options](#) 」プラグインの設定項目にある、フォーカス奪取防止ウィンドウ適用対象の指定欄の「 any 」を削除して空欄にすると、フォーカス奪取防止機能は無効になる。

特別なケース

あらゆるプログラムのあらゆるウィンドウが一貫した条件で識別できれば素晴らしいのだが、そのようになってはいないのが実状である。

様々なプログラムが様々な識別子を使用していて、そのなかには本当に奇妙なものがある。

`Gtk+` を使用したプログラムを例に挙げると、メニューとポップアップ・メニューとドロップダウン・メニューは同じではない。

`type=popupmenu`

と記述すると、右クリックしたときにポップアップするコンテキスト・メニュー **のみ** を表わし、

type=dropdownmenu

は、プログラムのメニューバーでドロップダウン表示されるメニュー **のみ** を表す。

KDE の通常 (normal) のウィンドウではないメニューやツールチップ、コンボ・ボックス、ドラッグ&ドロップ・アイコンその他ほとんどすべては、

type=unknown

で該当させることが可能である。

Firefox や Thunderbird の通常 (normal) のウィンドウではないメニューやツールチップ、その他ほとんどは、**(class=Firefox-bin | class=Thunderbird-bin) & override_redirect=1**

で該当させることが可能である。

Java ウィンドウはタイプが全くの不明。しかし幸いなことに、クラス名を利用すれば判別可能：

- 通常ウィンドウ向け: **name=sun-awt-X11-XFramePeer for normal Java windows**
- メニューやツールチップ向け: **name=sun-awt-X11-XWindowPeer**
- ダイアログ・(サブ)ウィンドウ向け: **name=sun-awt-X11-XDialogPeer**

なお、KDE、Firefox と Thunderbird、Java の三点については上記のようにしなくても、「[次善策](#)」プラグインでオプションを有効にすれば、自動的に修正することが可能である。

参考：

- [Compiz-Fusion.org Wiki](#)
- [試験運用中なLinux備忘録](#)