

# Guida a wmii-3\*

Steffen  
Liebergeld

con l'aiuto di  
Salvador  
Peiró

July 12, 2006

## Contents

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Scopo del documento . . . . .	3
1.2	wmii—la seconda generazione del window manager improved . . . . .	3
1.3	A chi é rivolta questa guida . . . . .	3
<b>2</b>	<b>Configurazione e installazione</b>	<b>4</b>
2.1	Ottenere wmii . . . . .	4
2.2	Configurazione e installazione . . . . .	4
<b>3</b>	<b>Terminologia</b>	<b>5</b>
3.1	Client . . . . .	5
3.2	Focus . . . . .	5
3.3	Eventi . . . . .	5
3.4	Tag . . . . .	5
3.5	Viste . . . . .	6
3.6	Colonne . . . . .	6
3.7	Layout . . . . .	6
<b>4</b>	<b>Come iniziare</b>	<b>6</b>
4.1	Primi passi . . . . .	7
4.2	Usare le colonne . . . . .	7
4.3	E i layout? . . . . .	8
4.4	Livello fluttuante . . . . .	8
4.5	Tag . . . . .	8
4.6	Come faccio a chiudere una finestra? . . . . .	9
4.7	Come avvio un programma? . . . . .	9
4.8	Come esco da wmii? . . . . .	9

---

\*Grazie a tutta la comunità di wmii, in particolare tutte le persone citate in [WMII/people](#).

<b>5</b>	<b>Cosa c'è sotto il cofano</b>	<b>9</b>
5.1	Window management dinamico . . . . .	9
5.2	Modularità—strumenti diversi per compiti diversi . . . . .	9
5.3	Il collante che tiene tutto insieme —9P . . . . .	10
5.4	Strumenti . . . . .	10
5.5	Conclusione . . . . .	11
<b>6</b>	<b>Scripting in wmii</b>	<b>11</b>
6.1	Linguaggio . . . . .	11
6.2	Azioni . . . . .	11
6.3	wmiirc . . . . .	12
6.4	Cambiare lo stile . . . . .	12
6.5	Usare la barra di stato . . . . .	13
6.6	Assegnare nuovi tag . . . . .	14
<b>7</b>	<b>Fine</b>	<b>14</b>
<b>8</b>	<b>Appendice</b>	<b>15</b>
8.1	filesystem . . . . .	15
8.2	combinazioni di tasti . . . . .	17
<b>9</b>	<b>Credits</b>	<b>18</b>
<b>10</b>	<b>Copyright</b>	<b>18</b>

# 1 Introduzione

## 1.1 Scopo del documento

Questa guida vuole essere un buon punto di partenza per i neofiti di *wmii-3*. Coloro che hanno già usato *wmi*, *wmii-2.5* o anche *ion* avranno modo di apprezzare le novità e le differenze di *wmii-3*. Chi invece non ha mai utilizzato un window manager a schede si innamorerà di questo nuovo sistema.

## 1.2 *wmii*—la seconda generazione del window manager improved

*wmii-3* è un nuovo tipo di window manager. È progettato per utilizzare poca memoria, per essere modularizzato all'estremo e per essere scritto con il minor numero possibile di righe di codice, garantendo cosa il minor numero possibile di bug. Infatti, uno dei nostri scopi ufficiali è quello di non superare le 10k righe di codice <sup>1</sup>.

*wmii* vuole essere facilmente portabile attraverso le piattaforme, dando così massima libertà all'utente.

*wmii-3* è la terza major release della seconda generazione del window manager improved <sup>2</sup>. Con la versione 2.5 di *wmii* è stato introdotto per la prima volta il concetto di window manager dinamico, tale da superare le limitazioni del paradigma WIMP (vedi anche *wmii.tex*).

## 1.3 A chi é rivolta questa guida

Daremo per scontato che il lettore abbia già una certa esperienza di Unix, ne conosca la terminologia base e i concetti di file e di editor.

Ci auguriamo che siate aperti a nuove idee, e che siate disponibili a investire un po' di tempo a imparare *wmii-3* <sup>3</sup>.

Se volete solo sapere come usare *wmii-3* e non siete interessati al suo funzionamento interno o allo scripting, potete limitarvi a leggere le sezioni 2, 3, la sottosezione ?? e saltare il resto.

Ad ogni modo, per ottenere il massimo da *wmii-3* dovrete leggere l'intero documento in maniera "sequenziale", dall'inizio alla fine. Un'altra possibilità è quella di leggere/usare la guida "on demand", man mano che vi rendete conto di aver bisogno di maggiori informazioni o volete capire meglio alcuni concetti. Raccomandiamo di leggere prima i capitoli introduttivi, per prendere confidenza con il mondo di *wmii*, e leggere i capitoli sullo scripting in un secondo momento.

---

<sup>1</sup> un beneficio di questo limite è che il codice è più facile da leggere e comprendere, e di conseguenza da usare e conoscere bene

<sup>2</sup> le due ii stanno infatti per il numero romano 2.

<sup>3</sup> ricordate l'adagio: "nessuno ti può insegnare quello che non vuoi imparare".

## 2 Configurazione e installazione

### 2.1 Ottenere *wmii*

*wmii* è rilasciato sotto licenza MIT/X Consortium License, quindi fondamentalmente si tratta di software libero, che potete scaricare gratuitamente da <http://wmii.de><sup>4</sup>.

### 2.2 Configurazione e installazione

Prima di tutto, controllate che non esistano pacchetti binari di *wmii* per la vostra distribuzione. Debian, Ubuntu e Gentoo dovrebbero avere già dei pacchetti pronti. Se avete a disposizione dei pacchetti di cui vi fidate, potete saltare tranquillamente alla prossima sezione.

Per tutti quelli che stanno ancora leggendo, sappiate che siete comunque avvantaggiati, perché compilando voi stessi i sorgenti, avrete tutti i file nelle posizioni originali, il che renderà più facile l'utilizzo di *wmii*.

1. Per disinstallare una versione precedente:

```
cd /percoroso/per/wmii-vecchio
make uninstall && make clean
```

Nel caso in cui stiate installando una nuova versione di *wmii*, questa è la prima cosa da fare, o finirete per mescolare versioni potenzialmente incompatibili dei file binari, dei file di configurazione e delle pagine di manuale.

2. Scompattate i sorgenti:

```
tar xzf wmii-3.tar.gz
cd wmii-3
```

3. Modificate la configurazione:

```
vim config.mk
```

La variabile più importante da valorizzare è `PREFIX` che imposta la directory di installazione di *wmii-3*. In caso di dubbio, lasciate il valore di default, non farete alcun danno al sistema.

4. Eseguite `make` e `make install`:

```
make && make install
```

---

<sup>4</sup> vedi <http://wmii.de/repos/wmii/LICENSE> per i dettagli

5. Configurate X-server per eseguire *wmii* come window manager predefinito. Potete farlo editando il file `~/.xinitrc`.

```
#!/bin/sh
exec wmi
```

Avete finito. Notate come gli autoconf tools non siano usati, per varie ragioni<sup>5</sup>. Non chiedete agli sviluppatori di *wmii* di usare autoconf, non sarete ascoltati.

## 3 Terminologia

Prima che muoviate i vostri primi passi in *wmii*, è necessario chiarire alcuni termini.

### 3.1 Client

Un client è un programma che vi presenta una interfaccia grafica per un utilizzo particolare, ad esempio un browser web o un terminale.

### 3.2 Focus

Il client che ha il focus è il client che riceve un vostro input. In X11 solo un client per volta può ricevere l'input. Se inserite dei comandi al terminale, la finestra che contiene detto terminale ha il focus, mentre tutte le altre finestre non ricevono l'input da voi inserito.

### 3.3 Eventi

Un evento è un messaggio generato dal server X per segnalare ai client X degli stati precisi. Ad esempio, il server X genera un evento di pressione pulsante se fate click in una finestra.

### 3.4 Tag

Un tag (o etichetta) è una stringa alfanumerica che potete associare ai client, e che vi permette di raggruppare i client in modo naturale.

In *wmii* non ci sono più workspace. Al loro posto, vengono visualizzati tutti i client che hanno un certo tag. Per esempio, se etichettate il vostro browser e un terminale con il tag "web-browser", e fate richiesta di visualizzare tutti i client che contengono questo tag, *wmii* mostrerà a video il browser e il terminale. È anche possibile assegnare ai client più di un tag, come descritto più avanti.

---

<sup>5</sup> leggete <http://www.ohse.de/uwe/articles/aal.html> e <http://lists.cse.psu.edu/archives/9fans/2003-November/029714.html> per ulteriori dettagli

## 3.5 Viste

Una vista (o *view*) è un gruppo di client che contengono un tag specifico e sono visualizzati. Una vista è abbastanza simile alla metafora del “workspace” in altri window manager, ma è più potente.

Può essere visibile solo una vista per volta.

Le viste sono in relazione con i tag che sono in uso in un dato momento. Esiste esattamente una vista per ogni singolo tag, quindi potete visualizzare solo gruppi di client che hanno un tag esistente.

Se eliminate l'ultimo client con un dato tag, la vista relativa a questo tag è distrutta.

## 3.6 Colonne

Una colonna è una parte di una vista in cui i client sono disposti automaticamente in direzione verticale.

In *wmii* è possibile suddividere ogni vista in più colonne. Ogni colonna contiene almeno un client, quindi non appena chiudete l'ultimo client di una colonna, questa viene automaticamente eliminata.

## 3.7 Layout

Un layout è la disposizione dei client in una colonna. Ci sono tre modi diversi per disporre i client.

**default** Questo layout divide lo spazio in parti uguali tra tutti i client di una colonna.

**massimizzato** Questo layout assegna a tutti i client le stesse dimensioni della colonna, mostrando un solo client per volta.

**stack** Questo layout dispone i client uno sopra l'altro, rendendo visibile solo quello con il focus, mentre gli altri sono accessibili tramite la barra del titolo. Si tratta di un approccio alternativo alla disposizione a *schede* (o *tabbing*).

## 4 Come iniziare

Ora possiamo tuffarci nell'esperienza di *wmii*. Vi suggerisco di provare immediatamente ogni procedura descritta anziché leggere tutto d'un fiato, così da evitare “perdite di memoria”. Sarebbe utile stampare questo documento o tenerlo a disposizione su uno schermo diverso, perché potreste non essere in grado di raggiungerlo durante i vostri primi passi con *wmii*.

Nota bene: il tasto *MOD* a cui si fa riferimento può rappresentare cose diverse su sistemi diversi. Di default in X11 è il tasto *Mod1* o *Alt*. Di norma è segnato come *Alt* sulla tastiera.

La notazione *MOD-tasto* significa che bisogna premere *MOD* e il tasto *tasto* allo stesso tempo.

Si possono configurare liberamente tutte le combinazioni di tasti, ma per semplicità ci limiteremo in questa guida alle combinazioni predefinite. Vedrete come cambiare le impostazioni di default nella sezione 6.

Le combinazioni di default usano parecchio i tasti *h* (sinistra), *j* (giù), *k* (su), and *l* (destra), associate alle relative direzioni.

## 4.1 Primi passi

Avviate ora l'interfaccia grafica di X. Dal momento che è la prima volta che usate *wmii*, vedrete una finestra con un breve tutorial. Siete liberi di leggerlo, ma potreste anche seguire questa guida :-)

Prima di tutto, premete *MOD-Invio* per avviare un xterm. Quest'ultimo occuperà metà dello spazio verticale, così che avrete due finestre equamente distribuite nello schermo. Se premete *MOD-Invio* ancora una volta, vedrete tre finestre equamente disposte.

Per passare da una finestra all'altra, premete *MOD-j*, che assegna il focus via via alle tre finestre.

Potete anche premere *MOD-k* per passare alla finestra sopra o *MOD-j* per passare alla finestra sotto quella con il focus.

Ora osservate le barre del titolo delle tre finestre. Mostrano delle informazioni importanti: la prima parte contiene il tag della finestra; la seconda il titolo della stessa.

Informazioni analoghe vengono mostrate nella barra di stato al fondo dello schermo. Le prime etichette mostrano i tag in uso, con in evidenza la vista corrente. Sul lato destro ci sono delle informazioni sullo stato del sistema: di default l'ora corrente e il carico di cpu (vedi la sottosezione 6.5 per maggiori dettagli).

## 4.2 Usare le colonne

Come descritto prima, *wmii* usa delle colonne per disporre le finestre. La vista corrente consiste già di una singola colonna. Ora creerete una nuova colonna.

In *wmii* le colonne consistono sempre di almeno un client, quindi per creare una nuova colonna dovete avere almeno due client a disposizione.

Selezionate un client a scelta e premete *MOD-Shift-l*, che sposta il client verso destra. Come vedete, *wmii* crea una nuova colonna dividendo la vista orizzontalmente in due aree della stessa dimensione. Il client selezionato è stato spostato nella nuova colonna.

Se chiudete l'ultimo client di una colonna, questa è immediatamente distrutta. Se chiudete l'ultimo client della vista corrente, anche la vista è distrutta.

Se premete *MOD-j* per cambiare il focus, vedrete che *wmii* cambia effettivamente il focus solo all'interno della colonna.

Per spostare il focus in un'altra colonna, potete premere rispettivamente *MOD-l* (destra) e *MOD-h* (sinistra).

È anche possibile scambiare di colonna client adiacenti. Per scambiare un client verso sinistra, premete *MOD-Control-h*. Per effettuare uno scambio verso destra, premete *MOD-Control-l*.

### 4.3 E i layout?

I layout dispongono i client in una colonna, sono quindi relativi ad una singola colonna per volta. In questo modo è possibile avere più di una colonna per vista, ognuna con un layout differente.

Il layout di default dispone ogni client di una colonna dividendo equamente lo spazio. Potete abilitare esplicitamente questo layout premendo *MOD-d* (dove “d” sta pre default).

Il layout a stack (pila) può essere abilitato con *MOD-s*. Come potete vedere, ora un solo client occupa più spazio che può, mentre gli altri sono visibili solo attraverso la barra del titolo.. Potete passare da un client all’altro premendo *MOD-j*.

Il layout massimizzato *MOD-m* assegna ai client le stesse dimensioni della colonna. È quindi visibile solo il client con il focus, tutti gli altri si trovano al di sotto di quest’ultimo. Potete sempre passare da un client all’altro premendo *MOD-j*.

### 4.4 Livello fluttuante

Il cosiddetto “livello fluttuante” è usato per gestire i client nella maniera classica, simile a quella dei window manager convenzionali. Alcuni client, infatti, non si adattano bene al sistema a schede, perché sono stati progettati avendo in mente i window manager convenzionali. Un esempio possono essere The Gimp e Xmms.

Anche se *wmii* è un window manager dinamico, che ordina automaticamente le finestre, questi programmi vecchio stile si appoggiano ad un concetto convenzionale di gestione delle finestre, in cui tutti i client fluttuano sul desktop e voi siete costantemente impegnati a mettere ordine.

Per assegnare questi client difettosi al livello fluttuante, potete alternare il focus tra il livello fluttuante e quello gestito automaticamente premendo *MOD-Spazio*. La combinazione di tasti *MOD-Shift-Spazio* alterna la finestra con il focus tra il livello fluttuante e quello automatico.

Nota bene: il livello fluttuante è identificato internamente come colonna zero.

### 4.5 Tag

Fino ad ora tutti i client avevano il tag “1”, ed era disponibile un’unica vista. Una sola vista, tuttavia, non è facilmente gestibile, una volta che si sono aperti diversi client che fanno cose diverse. Potreste invece volere una vista per ogni ambito, ad esempio una con l’editor e gli strumenti di programmazione, una con il browser e una terza con il jukebox musicale.

La buona notizia è che il concetto di tag (etichette) fornisce un sistema molto dinamico per ottenere questo tipo di raggruppamento.

Potete assegnare un altro tag al client con il focus premendo *MOD-Shift-Numero* dove Numero sta per una cifra da 0 a 9.

Potete passare da una vista all’altra premendo *MOD-Numero*.

Normalmente, ogni volta che avviate un nuovo client, questo eredita il tag della vista corrente.

Nel prossimo capitolo illustreremo modi più potenti di usare i tag. Imparerete ad assegnare tag multipli ad un client e ad usare delle stringhe come tag.

## 4.6 Come faccio a chiudere una finestra?

La maggior parte dei client di X hanno un'opzione di menu o un bottone per la chiusura dell'applicazione. Nei rari casi in cui non esistesse una soluzione a portata di mouse, come nella maggior parte dei terminali, potete premere *MOD-Shift-c* per chiudere una finestra.

## 4.7 Come avvio un programma?

Potete farlo da un terminale. Ma *wmii* possiede un menu dei programmi gestito da tastiera, a cui potete accedere premendo *MOD-p*. I contenuti di questo menu sono gestiti da un semplice shell script.

Vedrete dunque una lista di programmi. Se iniziate a digitare qualcosa, il menu accorcerà la lista, mostrando solo le voci che corrispondono alle lettere da voi inserite (nell'ordine in cui le avete inserite). Ogni volta che rimane una sola voce, il menu la evidenzia e voi potete avviare l'applicazione premendo *Invio*. Potete annullare l'operazione premendo *Esc*.

In questo modo, se volete lanciare Firefox, vi basterà digitare "fire" e premere *Invio*<sup>6</sup>.

## 4.8 Come esco da *wmii*?

Potete uscire da *wmii* tramite il menu azione (*MOD-a* e selezionando "quit". È tutto.

# 5 Cosa c'è sotto il cofano

In questo capitolo vedremo come è stato progettato *wmii*, quali idee sono state seguite dagli sviluppatori e come è stato implementato.

## 5.1 Window management dinamico

*wmii* è stato progettato intorno alla nuova idea di window management dinamico (gestione dinamica delle finestre). Questo significa che il window manager dovrebbe prendere tutte le decisioni circa la posizione delle finestre, togliendo un sacco di lavoro all'utente e permettendogli di concentrarsi sul suo lavoro. Il tutto può anche essere inteso come gestione tacita delle finestre.

## 5.2 Modularità—strumenti diversi per compiti diversi

Gli sviluppatori di *wmii* conoscono le idee più potenti che stanno alla base di Unix. Una di queste è l'idea di usare strumenti diversi per compiti diversi. Progettando accuratamente il window manager, sono stati in grado di dividere il tutto in programmi più piccoli, ognuno con un compito ben distinto dagli altri.

---

<sup>6</sup>Nel mio sistema è sufficiente digitare "efo" per avviare il browser ;-))

## 5.3 Il collante che tiene tutto insieme —9P

I programmi del mondo Unix comunicano, di norma, attraverso dei buffer accessibili tramite descrittori (di file), tra cui i socket.

Allo scopo di creare un protocollo di comunicazione leggero ma potente, gli sviluppatori di *wmii* hanno osservato attentamente il design di Plan9 e hanno scelto il protocollo 9P.

Le idee di base per la configurazione ed esecuzione di *wmii* sono state prese dall'interfaccia utente Acme per i programmatori di Plan9. Allo stesso modo di Acme, *wmii* fornisce un'interfaccia basata su filesystem, accessibile ai client 9P. Questo permette l'interazione di qualsiasi tipo di applicazione tramite filesystem, implementabile in qualsiasi linguaggio di programmazione si voglia. Questo svincola i programmatori di client da uno specifico linguaggio o paradigma.

L'interfaccia di *wmii* può essere comparata al filesystem *procfs* del kernel Linux.

Se volete interagire con un processo *wmii* in esecuzione, potete accedere al suo filesystem 9P usando lo strumento incluso *wmiir* o caricando il modulo 9P2000 per i kernel Linux a partire dal 2.4.16. Con il modulo del kernel avrete il vantaggio di poter montare il filesystem di *wmii* direttamente nella gerarchia di filesystem principale, ma ha lo svantaggio di richiedere privilegi di root.

## 5.4 Strumenti

Questa sezione intende fornire una panoramica sugli strumenti inclusi in *wmii*. Per una descrizione più dettagliata dovrete leggere le relative pagine man.

*wmiir* è un piccolo strumento usato per accedere al filesystem virtuale di *wmii* da remoto. Fondamentalmente supporta le seguenti operazioni:

- read (leggi)
- write (scrivi)
- remove (elimina)
- create (crea)

*wmiir* ha bisogno di conoscere l'indirizzo del filesystem con cui deve lavorare. All'avvio, *wmii* esporta la variabile d'ambiente `WMII_ADDRESS` che punta all'indirizzo del filesystem service di *wmii*. Questo può essere:

- un unix socket locale, del tipo `unix!/path/to/socket`
- un indirizzo di socket tcp, del tipo `tcp!hostname:port`

Se avete intenzione di lavorare su un filesystem diverso, potete specificarlo manualmente tramite l'opzione `-a indirizzo` da riga di comando. Un esempio dell'uso di *wmiir* potrebbe essere:

```
wmiir read /
```

Questo comando stampa il contenuto della directory root del filesystem virtuale di *wmii*.

*wmiimenu* è un menu generico comandato da tastiera, che seleziona le voci tramite delle sottostringhe. Potete saperne di più leggendo la pagina *man*.

*wmiiwarp* è un minuscolo strumento che sposta il puntatore del mouse verso coordinate specifiche dello schermo.

*wmiiwm* è il programma principale del window manager. Potete interagire con esso solo tramite il filesystem virtuale.

*wmiipsef* stampa il contenuto della selezione X corrente su STDOUT. Utile per gli script.

## 5.5 Conclusione

Il servizio di filesystem virtuale di *wmii* e gli strumenti qui presentati vi permettono di avere il controllo completo del window manager anche attraverso degli script. Vedremo degli esempi nella prossima sezione [6](#).

# 6 Scripting in *wmii*

In questo capitolo vedremo come controllare *wmii* attraverso degli script. Vi daremo le indicazioni di base così che possiate procedere autonomamente.

## 6.1 Linguaggio

Come già accennato, l'unico vincolo per interagire con *wmii* è la possibilità di accedere al suo filesystem virtuale. Il modo più semplice per farlo è tramite il programma *wmiir*. Quindi gli script di shell rappresentano la via più facile per adattare *wmii* alle vostre esigenze.

In realtà potete controllare *wmii* in qualsiasi linguaggio di programmazione. Ad ogni modo, gli script di default di *wmii* sono scritti usando un sottoinsieme di “sh”, quindi in piena osservanza POSIX, per rendere *wmii* più *portabile* possibile.

Per semplicità gli esempi seguenti faranno uso anch'essi di “sh”, e non richiederanno la conoscenza di python, tcl, ruby, ...

## 6.2 Azioni

In *wmii* potete raggruppare certi compiti in *azioni*. Le azioni non sono altro che semplici script che si trovano o nella directory di configurazione di default di *wmii*, o nella vostra. <sup>7</sup>

Premendo *MOD-a* attiverete il menu azione, che funziona in modo simile al menu programmi, ma mostra solo le azioni disponibili.

Se volete aggiungere le vostre azioni, potete inserire i vostri shell script nella directory di configurazione di default o in `$HOME/.wmii-3`.

---

<sup>7</sup> `$CONFPREFIX` è impostato in *config.mk* e di default punta a `/usr/local/etc` o a `$HOME/.wmii-3`

Il tutto funziona perché lo script di controllo di *wmii* esporta la variabile `PATH` come `PATH=~/.wmii-3:\$CONFPREFIX/wmii:\$PATH` prima di eseguire `wmiiwm`. In questo modo le azioni locali che si trovano in `~/.wmii-3` hanno la precedenza su quelle di default che si trovano in `\$CONFPREFIX/wmii`.

Potete modificare al volo questo file, non è necessario fermare *wmii*. Una volta finito, vi basterà eseguire `wmiiirc` e i cambiamenti saranno di effetto immediato. Per fare questo, aprite il menu azione (premendo *MOD-a* e scegliete *wmiiirc*). È anche possibile eseguire azioni di *wmii* da un terminale, utile effetto collaterale dell'esportazione della variabile `\$PATH` nello script di avvio di *wmii*.

### 6.3 wmiirc

Si tratta di uno speciale script in “sh” che viene eseguito all'avvio di *wmii* per la configurazione e il controllo dello stesso.

Per ottenere questo effetto scrive diversi file nel filesystem virtuale di *wmii* e legge gli eventi segnalati da *wmii* durante l'esecuzione. Gli eventi consistono soprattutto di combinazioni da tastiera, click del mouse o definiti dall'utente. Gli eventi sono gestiti da un'iterazione nello script.

Così per una configurazione base di *wmii*, come ad esempio cambiare il tasto *Mod1* o i tasti di navigazione, vi potete rivolgere a questo script.

Il nome *wmiiirc* significa *wmii run command*, perché “rc” è una vecchia abbreviazione Unix di “run command”.

### 6.4 Cambiare lo stile

Lo stile di *wmii-3* è definito attraverso font e colori, esportati tramite le seguenti *variabili d'ambiente*.

```
WMII_SELCOLORS='#000000 #eaffff #8888cc'  
WMII_NORMCOLORS='#000000 #ffffea #bdb76b'  
WMII_FONT=static
```

`WMII_SELCOLORS` definisce i colori del bordo e della barra del titolo della finestra attiva, `WMII_NORMCOLORS` definisce i colori del bordo e della barra del titolo dei client non selezionati. I numeri sono valori rgb in esadecimale, come forse avete già visto nel linguaggio HTML. Potete selezionarli, ad esempio, tramite il selettore di colore di Gimp.

Il primo valore definisce il colore del testo delle stringhe nelle barre e nei menu. Il secondo valore definisce il colore di sfondo delle barre e dei titoli dei client, mentre il terzo definisce il colore del bordo di 1px che circonda le barre e i client.

`WMII_FONT` definisce il font che deve essere usato per visualizzare il testo nelle barre del titolo, nella barra di stato e nei menu.

Potete visualizzare i vari font usando, ad esempio, *xfontsel*.

## 6.5 Usare la barra di stato

La barra di stato in *wmii* ha la propria directory `bar` con una sottodirectory per ogni label (etichetta) che viene creata. Mentre editavo questo documento la mia barra di stato aveva questo aspetto:

```
$ wmiir read /bar
d-r-x----- salva salva      0 Mon Apr 17 14:19:51 2006 1
d-r-x----- salva salva      0 Mon Apr 17 14:19:51 2006 2
d-r-x----- salva salva      0 Mon Apr 17 14:19:51 2006 status
```

Allo stesso tempo ogni directory contiene due file:

```
$ wmiir read /bar/status
--rw----- salva salva      23 Mon Apr 17 14:22:14 2006 colors
--rw----- salva salva      23 Mon Apr 17 14:22:14 2006 data
```

Il primo file contiene le definizioni di colore che controllano il modo in cui la barra è disegnata, mentre il secondo contiene i dati che vengono mostrati.

Ora potete fare i vostri esperimenti creando una nuova label, esplorarla e modificarla leggendo e scrivendo valori diversi nei file `colors` e `data`. Una caratteristica interessante della barra (e dei client) è che generano eventi che corrispondono a click del mouse su di essi. Potete aprire un terminale ed eseguire `wmiir read /event` per vedere come gli eventi vengono generati quando fate click sulla barra di stato. Si tratta di un meccanismo che permette di controllare le applicazioni direttamente dalla barra. Se avete finito e volete sbarazzarvi della label, eseguite il comando `wmiir remove /bar/foo`.

Se volete saperne di più, date un'occhiata allo script di status, e visitate <http://wmii.de> per altri esempi interessanti, come i seguenti:

- *status*: Tiene sotto controllo il livello della batteria e della temperatura sui portatili
- *status-mpd*: Controlla il demone mpd
- *status-load*: Mostra il carico della cpu
- *status-net*: Controlla il segnale della rete wireless

Ora, aprite lo script di status di default e cercate di capire come funziona: . La prima riga consiste in una comoda dichiarazione di funzione, `xwrite`, che evita un bel po' di inutili ripetizioni in fase di scrittura su file. Le tre righe successive creano e inizializzano la label di `status`. Infine l'ultima sezione è un ciclo `while` che cerca di inserire nella barra il carico di cpu e la data corrente.

La parte difficile è che cerca di farlo. Che cosa potrebbe impedire allo script di aggiornare la barra? Se `xwrite` cercasse di scrivere in una label che non esiste (rimossa), fallirebbe, risolvendo il ciclo e uscendo dallo script. La cosa ha senso perché nessuno vuole scrivere su una label che non esiste.

Ora torniamo alle prime linee dello script, e notiamo come c'è uno `sleep delay` tra la rimozione della label e la sua creazione.

Questo garantisce che la label di `status` non esiste, così che qualunque altro script di `status` in esecuzione fallisca ed esca. In questo modo siamo sicuri che ci sarà sempre solo uno script di `status` in esecuzione, e manteniamo una corrispondenza biunivoca tra label e script.

## 6.6 Assegnare nuovi tag

Come accennato prima, è possibile ottenere risultati più interessanti con i tag rispetto ai comandi predefiniti da tastiera. Ad esempio, potreste voler usare una stringa come tag, o usare più di un tag per client. Per fare questo, dovrete separare i tag con un “+”.

```
echo -n web+code | wmiir write /view/sel/sel/tags
```

Questo comando assegna al client che ha il focus i tag “web” e “code”. Ora potete selezionare la vista “web” tramite il seguente comando:

```
echo -n view web | wmiir write /ctl
```

Durante lo sviluppo di *wmii-3*, divenne chiaro che si tratta di un’azione così comune che è stata creata un’apposita combinazione di tasti. Di default *MODt* visualizza un menu di scelta della vista, mentre *MOD-Shift-t* visualizza un menu che permette di assegnare nuovi tag al client selezionato.

## 7 Fine

Ci auguriamo che questa guida vi abbia aiutato a prendere confidenza con *wmii*. Se avete notato qualcosa di sbagliato, confuso o mancante in questo documento, non esitate a scriverci.

Potete trovare tutte le informazioni per contattarci qui: [mail diretta](#), [\[wmii\]](#) mailinglist, [#wmii](#) canale irc, o anche tramite segnali di fumo <sup>8</sup>.

Ricordate anche che *wmii* è stato scritto da persone di buon gusto, quindi la maggior parte delle decisioni che sono state prese sono il frutto di ragionamenti approfonditi. Se pensate che qualcosa non abbia senso o non si accordi con il quadro generale, cercate di capire da soli cosa sta succedendo prima di porre delle domande. Probabilmente finirete per imparare molte cose, e se alla fine dovrete davvero scoprire degli errori potrete fornirci molti più elementi per risolvere il problema.

---

<sup>8</sup> ma non chiedeteci aiuto su questo, vi dovrete arrangiare da soli ; -P.

## 8 Appendice

### 8.1 filesystem

- /bar**
- namespace della barra
  - per aggiungere una label (etichetta), create `/bar/label` (che genera in automatico i file `colors` e `data`).
  - per cancellarla, remove `/bar/label`
- /bar/label**
- ogni barra ha il proprio namespace organizzato in base alla label
  - una label può essere qualsiasi stringa
- /bar/label/data**
- i dati presenti nella label `label`
- /bar/label/colors**
- i colori della barra
- /client**
- il namespace dei client
- /client/n**
- Ogni client, inclusi quelli non mostrati nella vista corrente ha un namespace qui
  - `n` è un intero non negativo, i client sono numerati in ordine dal più vecchio
- /client/n/class**
- un file contenente le informazioni `class:instance` per il client `n`
- /client/n/ctl**
- file di controllo per il client `n`
  - comandi accettati:
    - kill (elimina il client)
    - sendto `area|prev|next|new`
    - swap `up|down|prev|next`
- /client/n/geom**
- la geometria della finestra del client `n`
  - consiste in tre interi separati da spazio (`x y larghezza altezza?`)
- /client/n/index** contiene l'indice del client nel namespace `/client`, in questo caso `n`
- /client/n/name**
- il nome del client `n` così com'è visto da *wmii* (visualizzato nella tagbar)
- /client/n/tags**
- una lista separata dal segno + di tutti i tag applicati al client `n`
- /ctl**
- il file di controllo e l'interfaccia di comando di *wmii*
  - comandi accettati:
    - quit
    - retag
    - view `tag`
- /def**
- /def/border** larghezza del bordo attorno ai client
- /def/font** il font usato da *wmii* (secondo la specifica `xlib`)
- /def/keys** una lista separata da newline delle combinazioni intercettate da *wmii*

**/def/rules** • una lista separata da newline di regole che specificano come applicare tag automatiche ai client

- il confronto è fatto con il valore di `class.instance` del client
- sintassi: `/$regex/ -> $tag` (tag può essere `~` per indicare il livello fluttuante)

**/def/colwidth** larghezza di una nuova colonna (in px)

**/def/colmode** modo di una nuova colonna

**/view** • visualizzazione di una collezione di client associati ad un tag specifico

**/view/area** • ogni area ha un proprio namespace nella vista corrente

- le aree sono numerate a partire da 0
- 0 è sempre l'area fluttuante, le altre sono colonne

**/view/area/ctl** • comandi accettati:

- `select client | prev | next`
- `client` si riferisce al numero di client relativo al numero ed età dei client in `area`

**/view/area/mode** il layout corrente dell'area: `equal`, `stack` o `max`

**/view/area/sel** il client selezionato all'interno dell'area

**/view/area/client** il namespace per ogni client in `area`

**/view/ctl** comandi accettati: `select area | prev | next | toggle`

**/view/sel** l'area selezionata nel workspace

**/view/tag** il tag comune a tutti i client nel workspace

## 8.2 combinazioni di tasti

Qui ci sono le combinazioni di default. \$MODKEY è un segnaposto, solitamente corrisponde Mod1 or Alt.

Combinazione	Azione
<b>Spostare il Focus</b>	
\$MODKEY-h	sposta il focus alla colonna precedente
\$MODKEY-l	sposta il focus alla colonna successiva
\$MODKEY-j	sposta il focus al prossimo client nella colonna
\$MODKEY-k	sposta il focus al client precedente nella colonna
\$MODKEY-space	passa da/alla colonna fluttuante 0
\$MODKEY-[0-9]	seleziona il tag/vista [0-9]
<b>Spostare i client</b>	
\$MODKEY-Control-h	scambia il client con l'ultimo client della colonna precedente
\$MODKEY-Control-l	scambia il client con l'ultimo client della colonna successiva
\$MODKEY-Control-j	sposta in basso il client nella colonna
\$MODKEY-Control-k	sposta in alto il client nella colonna
\$MODKEY-Shift-h	manda il client alla colonna precedente
\$MODKEY-Shift-l	manda il client alla colonna successiva
\$MODKEY-Shift-space	sposta il client a/dalla colonna fluttuante 0
\$MODKEY-Shift-[0-9]	manda il client al tag/vista [0-9]
<b>Layout</b>	
\$MODKEY-d	seleziona il layout di default
\$MODKEY-s	seleziona il layout a stack (pila)
\$MODKEY-m	seleziona il layout massimizzato
<b>Funzioni della barra menu</b>	
\$MODKEY-a	scegli un'azione dalla barra menu
\$MODKEY-p	scegli un programma dalla barra menu
\$MODKEY-t	scegli una vista dalla barra menu
\$MODKEY-Shift-t	assegna uno o più tag al client dalla barra menu
<b>Client e Applicazioni</b>	
\$MODKEY-Return	apri un terminale
\$MODKEY-Shift-c	chiudi il client

Table 1: Combinazioni tasti di default in *wmii*

## 9 Credits

**Autore** Steffen Liebergeld

**Principale critico e inquisitore** Salvador Peiró

**Traduzione italiana** Stefano K. Lee

**Helpers** Anselm Garbe

Denis Grelich

Ross Mohn

Neptun Florin

Jochen Schwartz

Adrian Ratnapala

## 10 Copyright

guide to wmii-3

Copyright (C) 2005, 2006 by Steffen Liebergeld, Salva Peiró

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 2

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.