

Linee guida per le interfacce grafiche (parte II)

Fabio Vitali



Introduzione

Oggi esaminiamo in breve:

- ◆ Le linee guida di progettazione delle interfacce Windows
- ◆ L'organizzazione e la progettazione di interfacce X Windows



Windows

- Mouse e tastiera
- Finestre
- Menu
- Controlli
- Finestre di dialogo
- L'interfaccia a penna



Mouse e tastiera

Il mouse ha due bottoni. Il primo è usato per le selezioni. Il secondo serve per azioni dipendenti dal contesto ed altre opzioni

Azioni del mouse

- ◆ Spostamento, Pressione, Click, Doppio click, Trascinamento, Doppio trascinamento

Azioni della tastiera

- ◆ Tasti di editing: Delete, Backspace, Ins
- ◆ Tasti toggle: Ins, Caps Lock, Num Lock, Scroll lock, F8, Maiuscole+F8
- ◆ Tasti modificatori: Maiuscole, CTRL, ALT
- ◆ Tasti di navigazione: Home, End, PG UP, PG DOWN, Frecce, TAB

Selezione su immagini, oggetti, testo



Finestre (1)

SDI (Single Document Interface)

- ◆ L'applicazione controlla solo una finestra all'interno del quale vengono visualizzati i dati ed avviene l'input.
- ◆ La finestra possiede controlli standard e anche il menu dell'applicazione.
- ◆ Essa permette l'apertura di un solo documento alla volta.
- ◆ Però la possibilità di attivare più istanze della stessa applicazione senza duplicazione dello spazio codice permette di avere finestre multiple.
- ◆ Sistemi di comunicazione tra processi permetteranno poi un controllo centralizzato di caratteristiche globali (condivisione degli stessi documenti, elenco delle finestre, etc.)



Finestre (2)

MDI (Multiple Document Interface)

- ◆ Ci sono due tipi di finestre: finestre di applicazione e finestre di documento.
- ◆ Le finestre di applicazione costituiscono lo scheletro visuale fondamentale dell'applicazione. Esistono metodi standard per aprirla, chiuderla, spostarla, ridimensionarla, iconificarla, scrollare al suo interno. La finestra di applicazione contiene il menu dell'applicazione.
- ◆ Le finestre di documento stanno all'interno della finestra di applicazione, e ne vengono clippate. Solo i controlli fondamentali vengono ripetuti nella finestra di documento.



Finestre (3)

MDI (Multiple Document Interface) - segue

- ◆ Una finestra di documento massimizzata (utilizzando un apposito comando sulla barra del titolo) occupa lo sfondo della finestra di applicazione oscurando le altre finestre e assumendo l'aspetto di un'applicazione SDI.
- ◆ Le finestre di documento NON hanno controlli, a parte il menu di controllo, i pulsanti di minimizzazione e massimizzazione e le scrollbar
- ◆ La barra di messaggio sta in fondo alla finestra di applicazione e contiene informazioni sui comandi selezionati dall'utente o per richiedere informazioni all'utente.
- ◆ Altre barre di controllo sono i righelli, i ribbon, le toolbox e le palette.



Menu (1)

Menu drop-down

- ◆ Menu dell'applicazione. Sta in cima alla finestra dell'applicazione, sotto la barra del titolo
- ◆ Cliccando su una voce, la barra rimane aperta finchè non viene premuto di nuovo da qualche altra parte il mouse o non viene selezionata una voce.
- ◆ Si può accedere ai menu tramite tastiera (in Macintosh si può solo attivare un comando noto)

Menu pop-up

- ◆ Appare in mezzo alla finestra quando invocati dall'utente, alla POSIZIONE DEL CURSORE.
- ◆ Contiene comandi dipendenti dal contesto.



Menu (2)

Menu a cascata (o gerarchico)

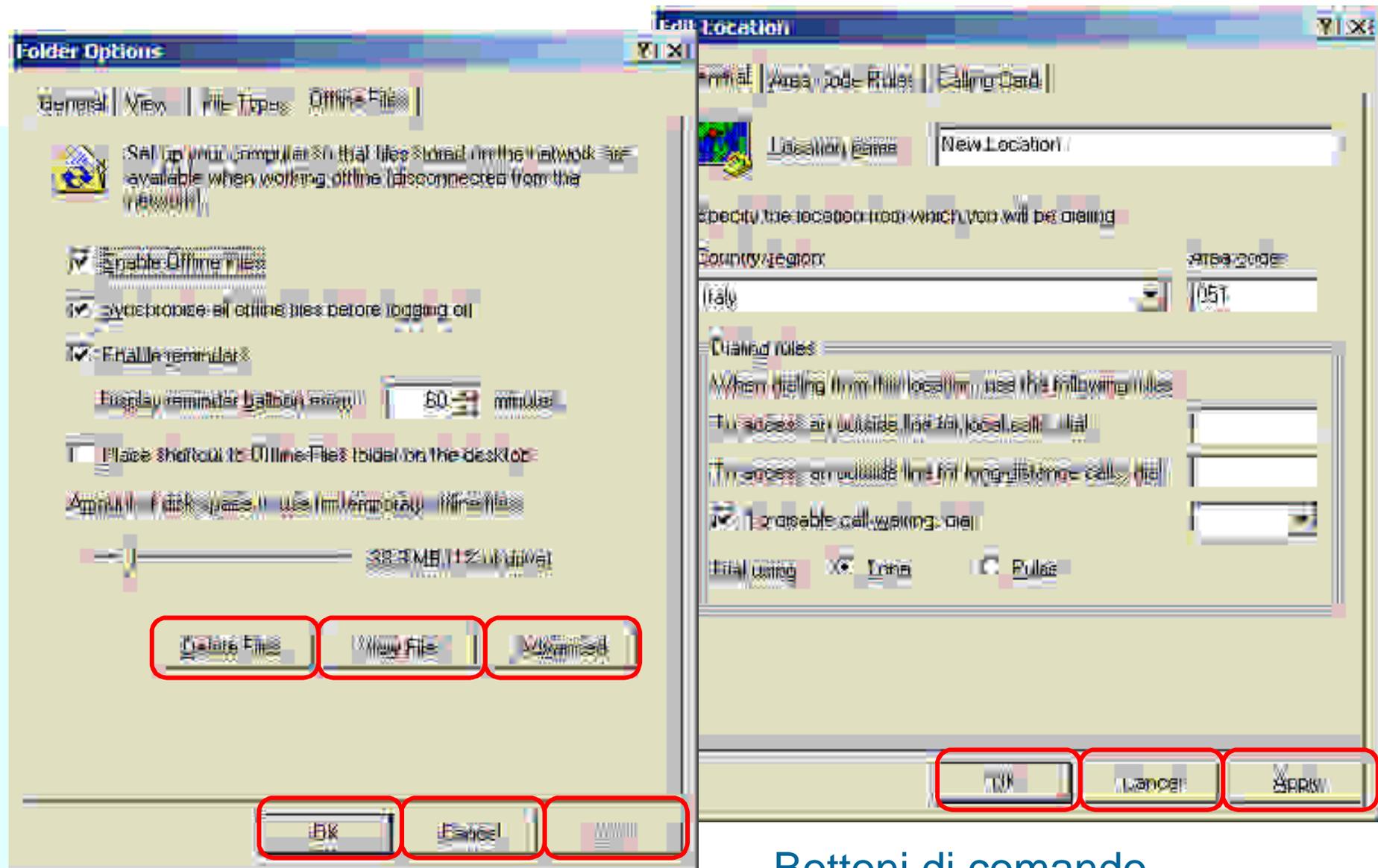
- ◆ Una voce di un menu può attivare un sottomenu a cascata, contenente ulteriori comandi.
- ◆ Si suggerisce di non usare più di due livello di menu.

Menu standard

- ◆ Menu control
- ◆ Menu file
- ◆ Menu edit
- ◆ Menu help
- ◆ View (opzionale)
- ◆ Windows (obbligatorio per MDI, opzionale per SDI)



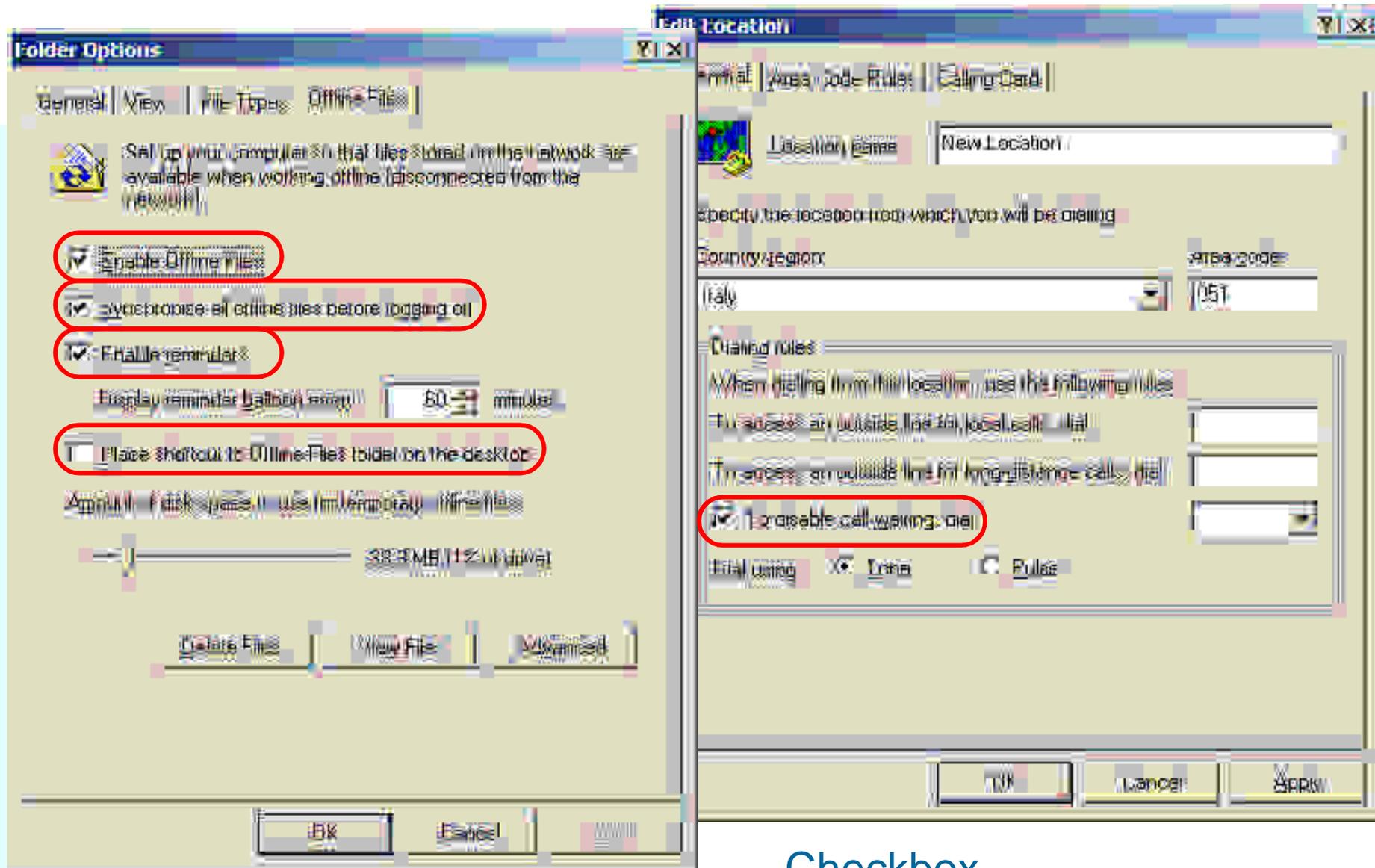
Controlli: bottoni (1)



Bottoni di comando



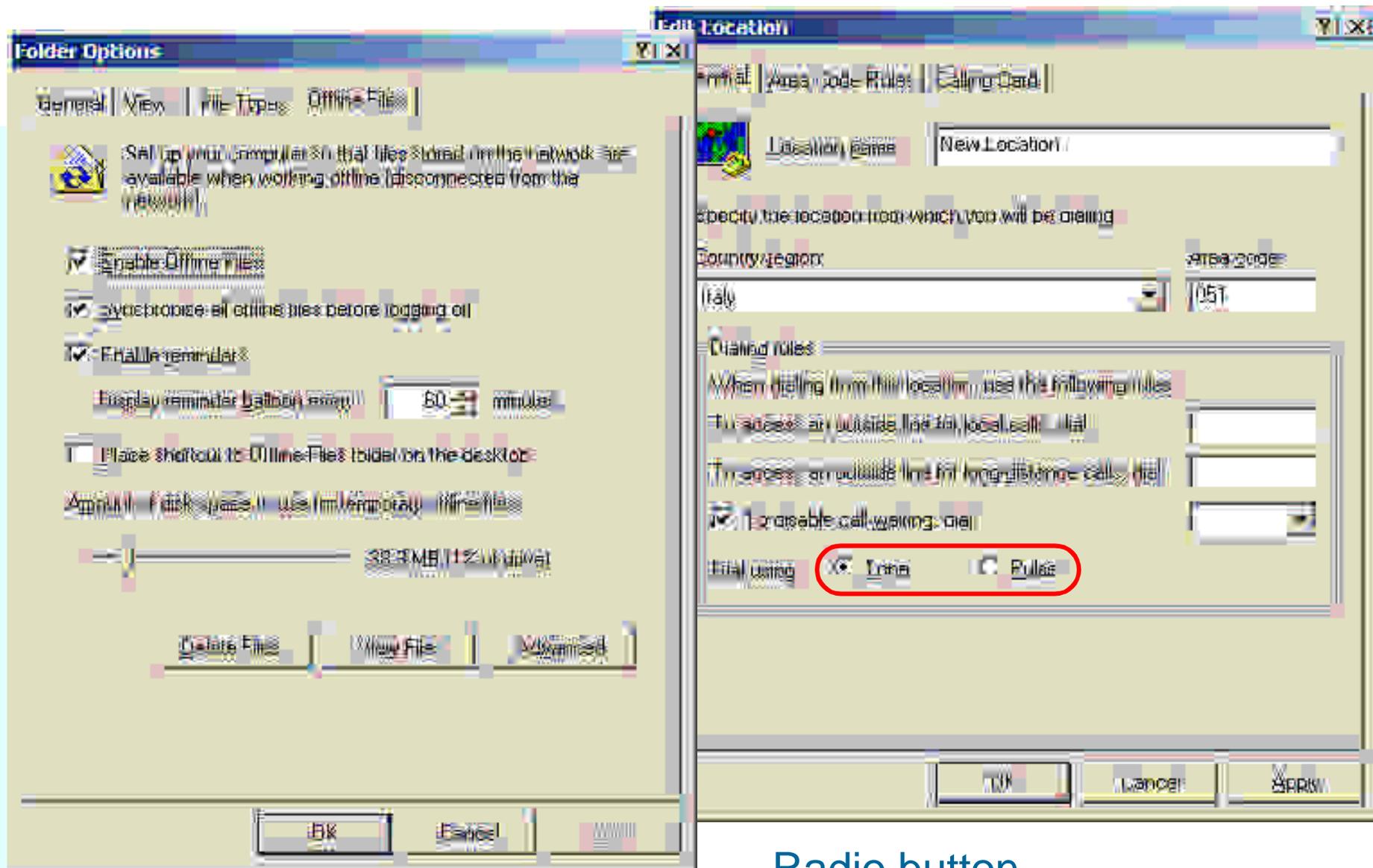
Controlli: bottoni (2)



Checkbox



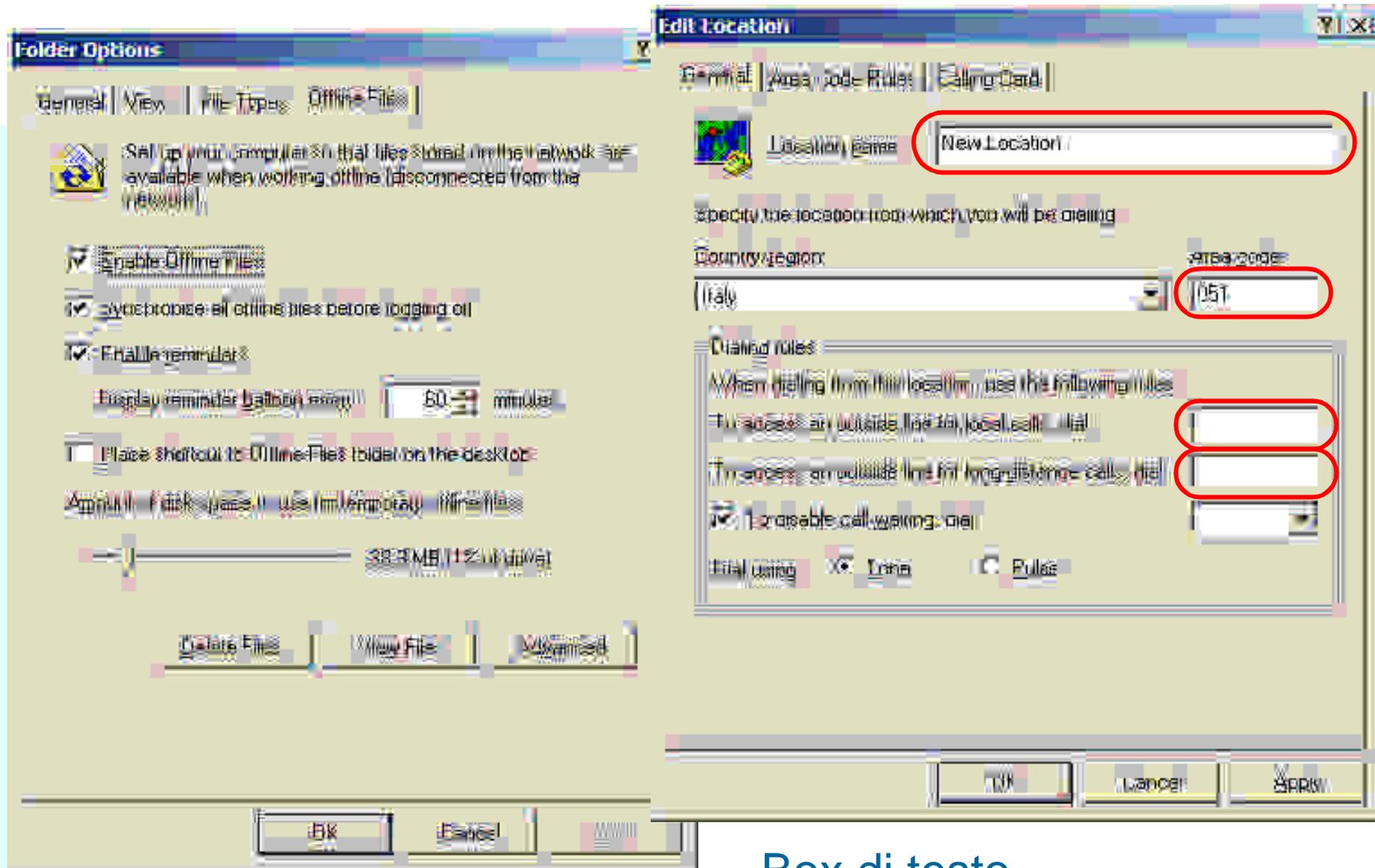
Controlli: bottoni (3)



Radio button



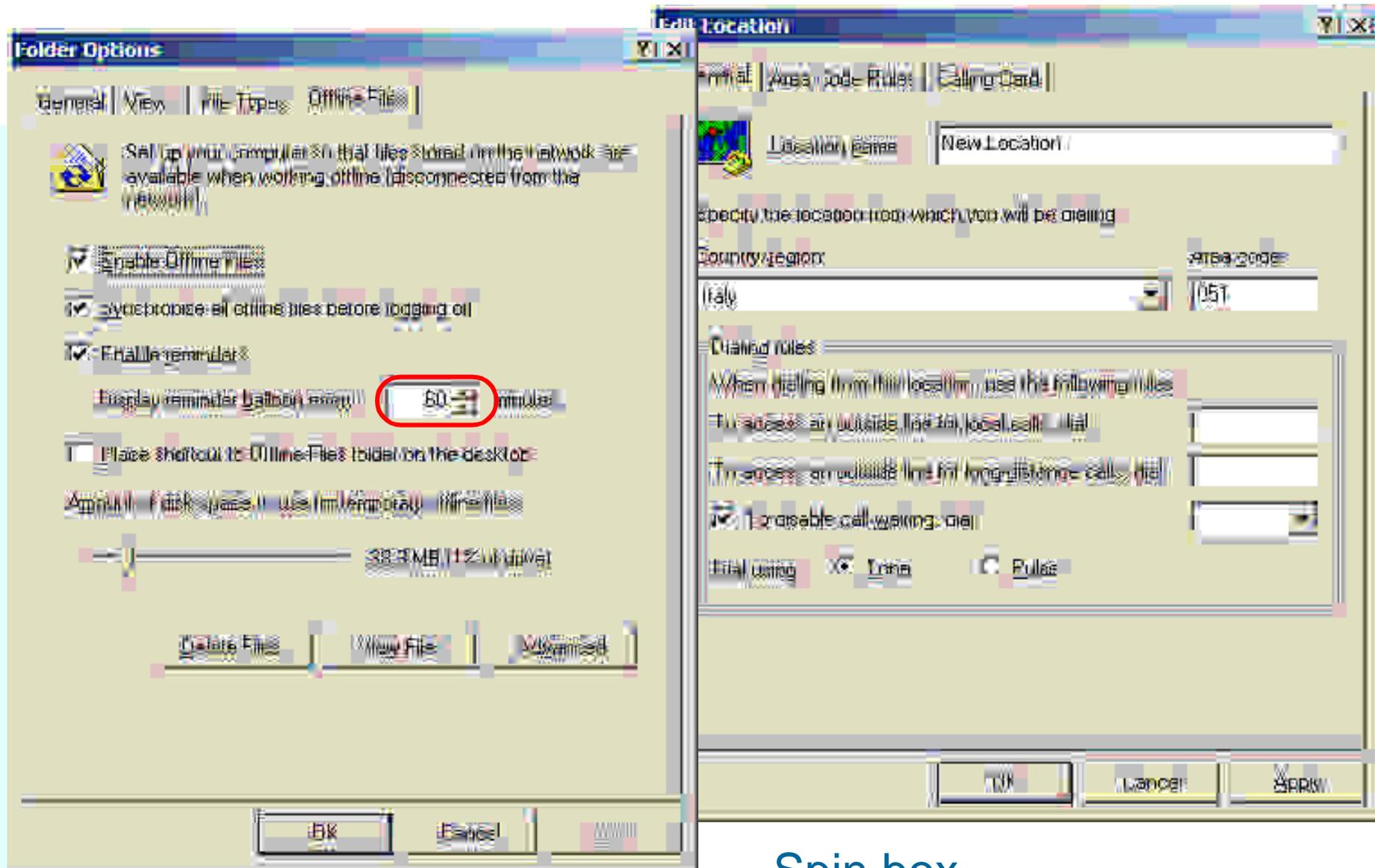
Controlli: testo (1)



Box di testo



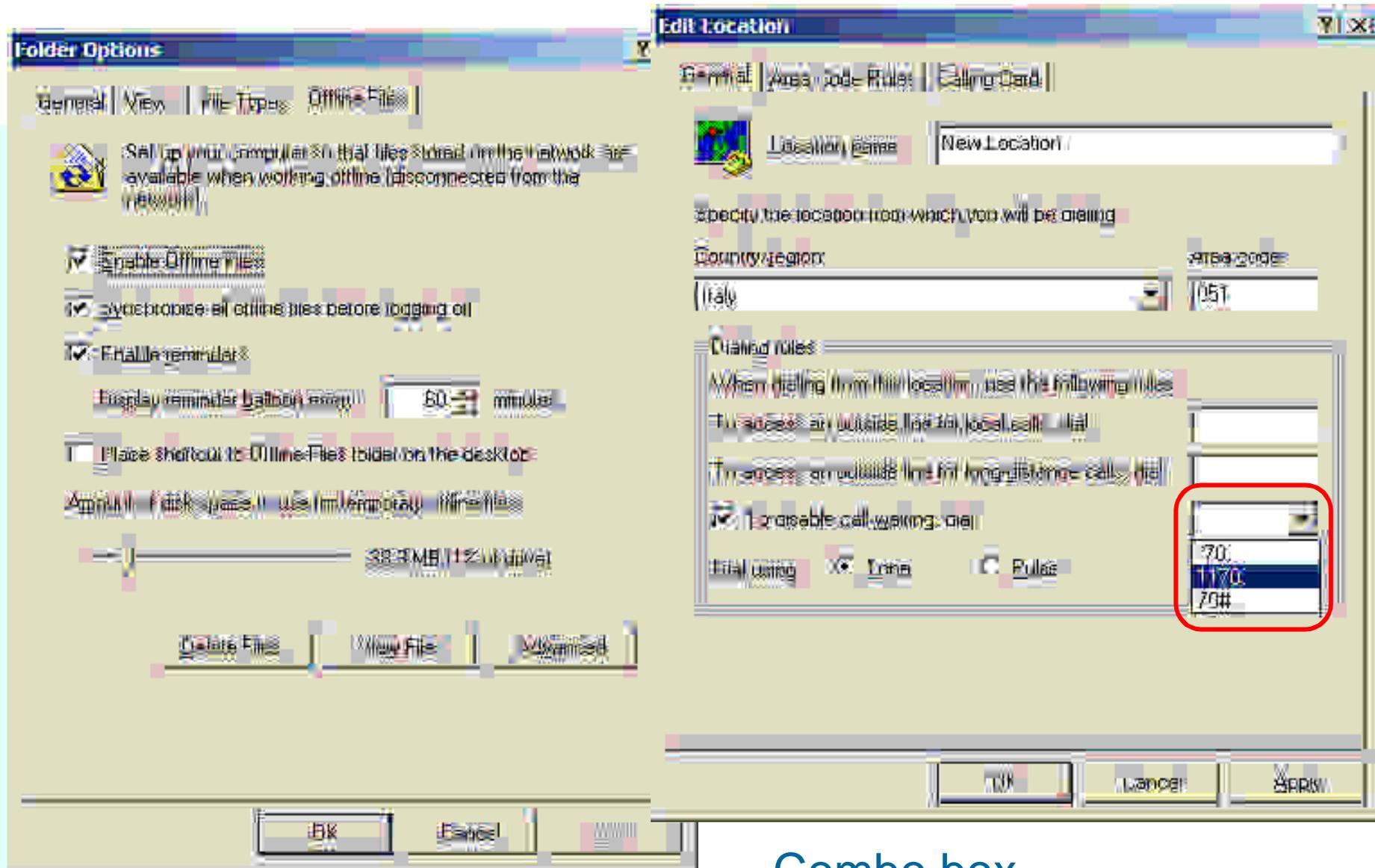
Controlli: testo (2)



Spin box



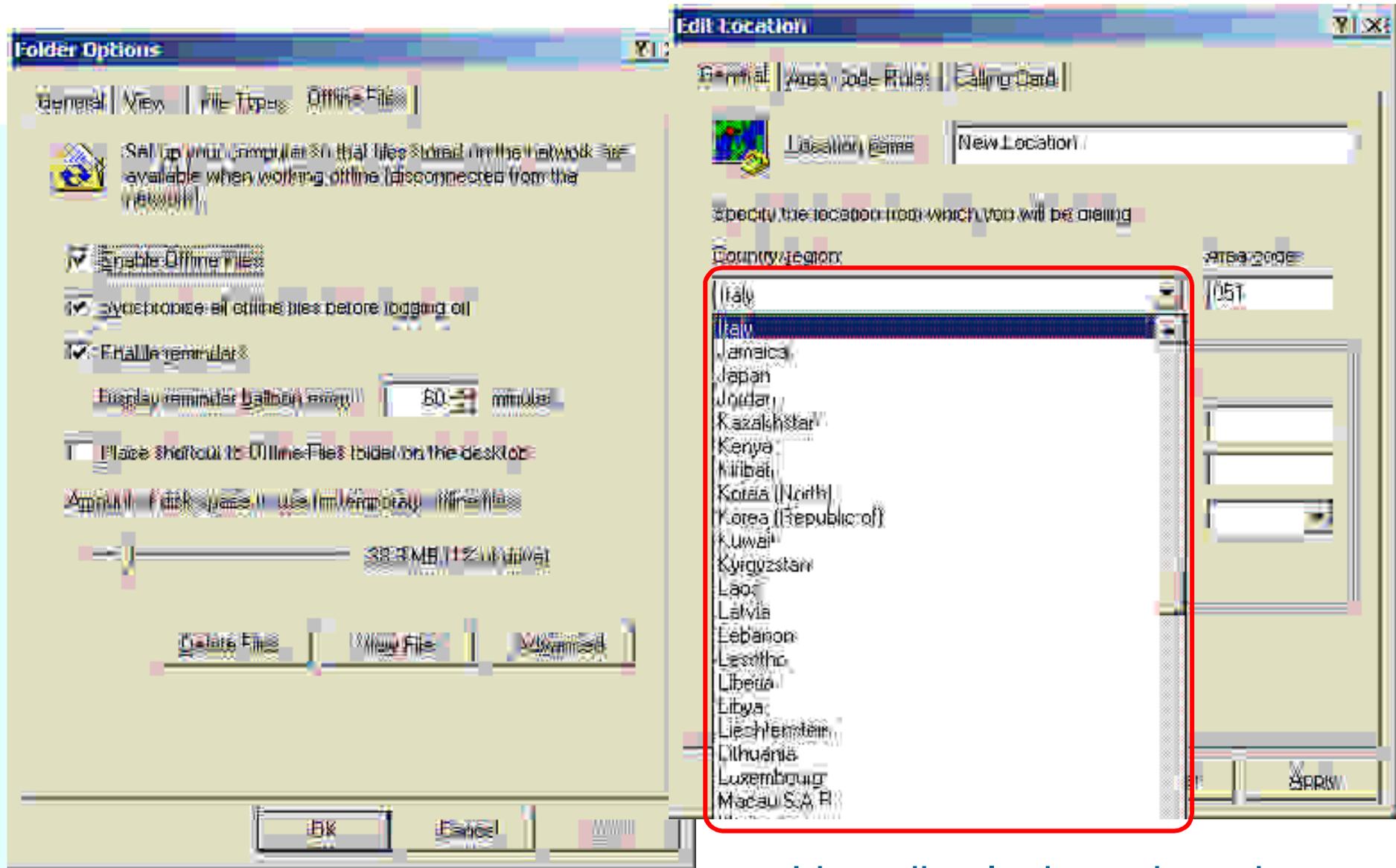
Controlli: testo (3)



Combo box



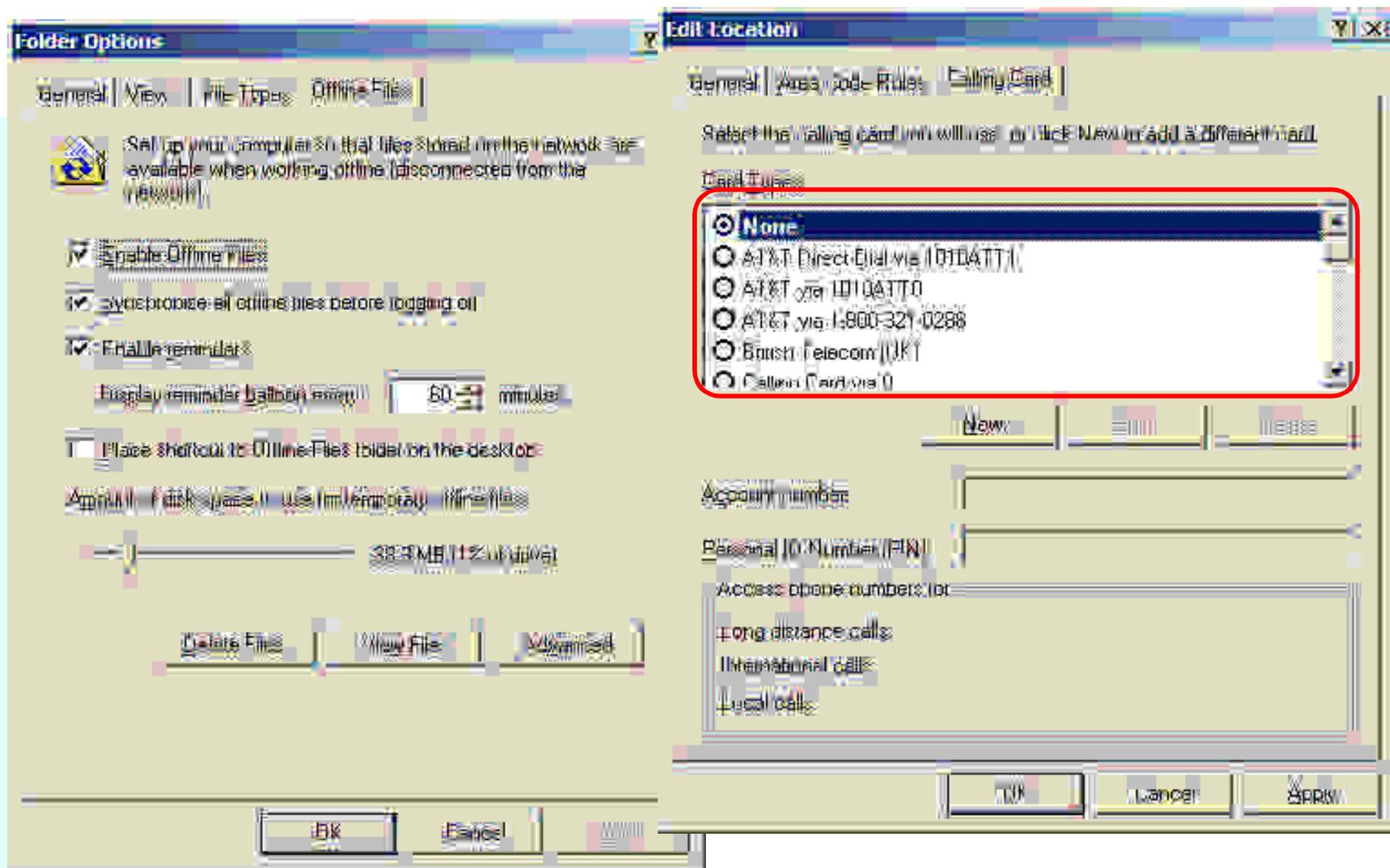
Controlli: lista (1)



Lista di selezione drop-down



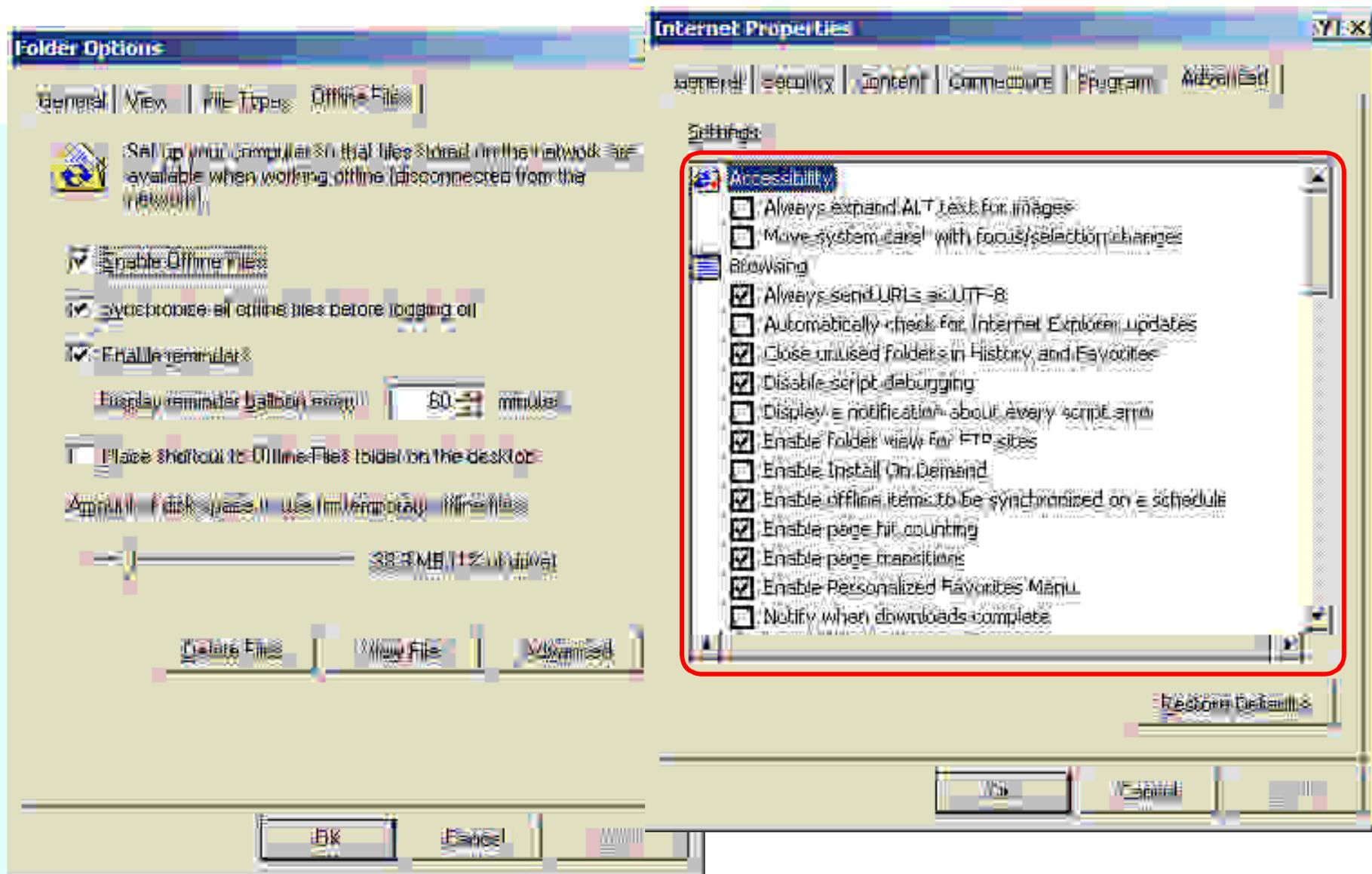
Controlli: lista (2)



Lista di selezione singola



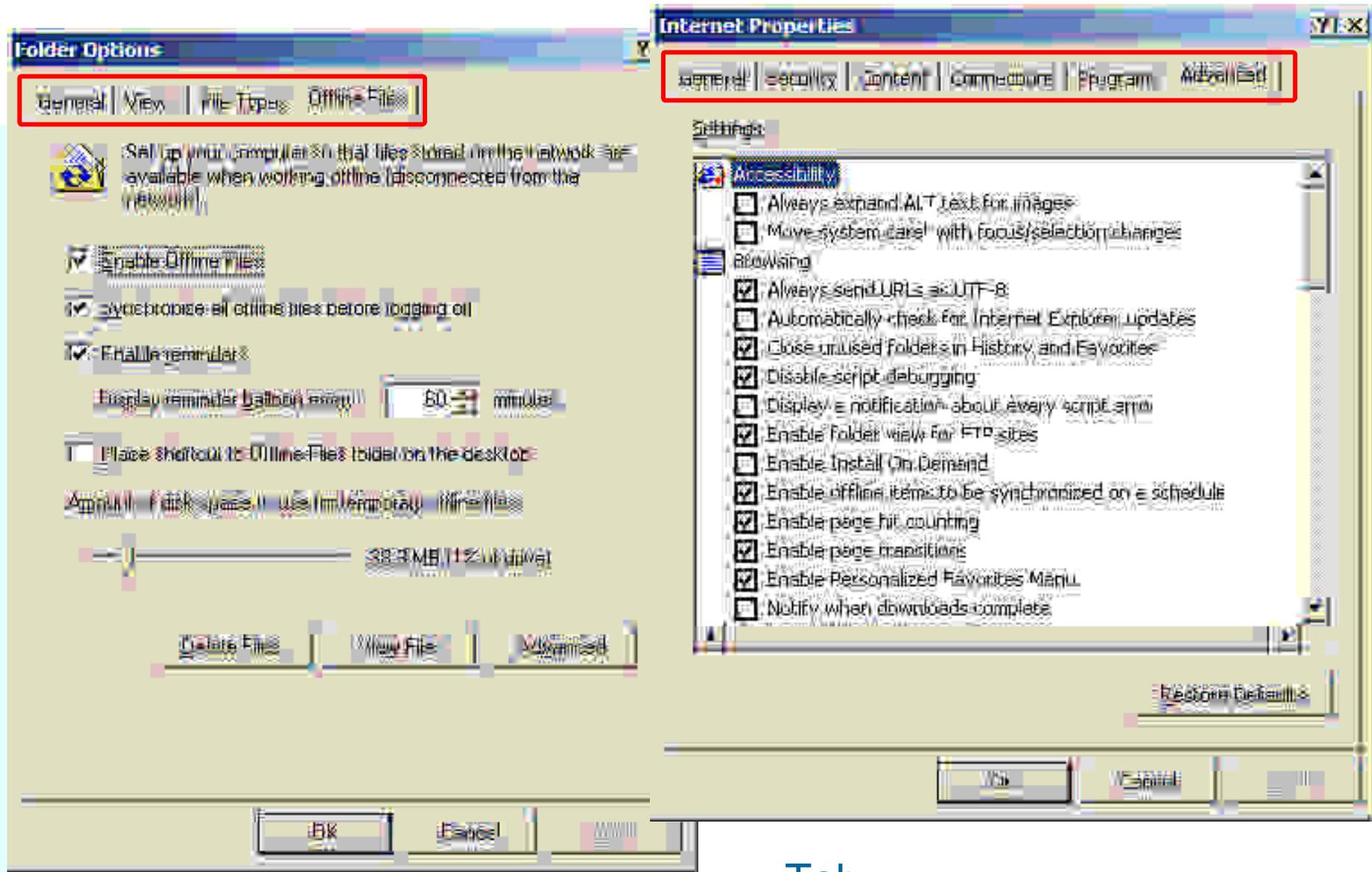
Controlli: lista (3)



Lista di selezione multipla



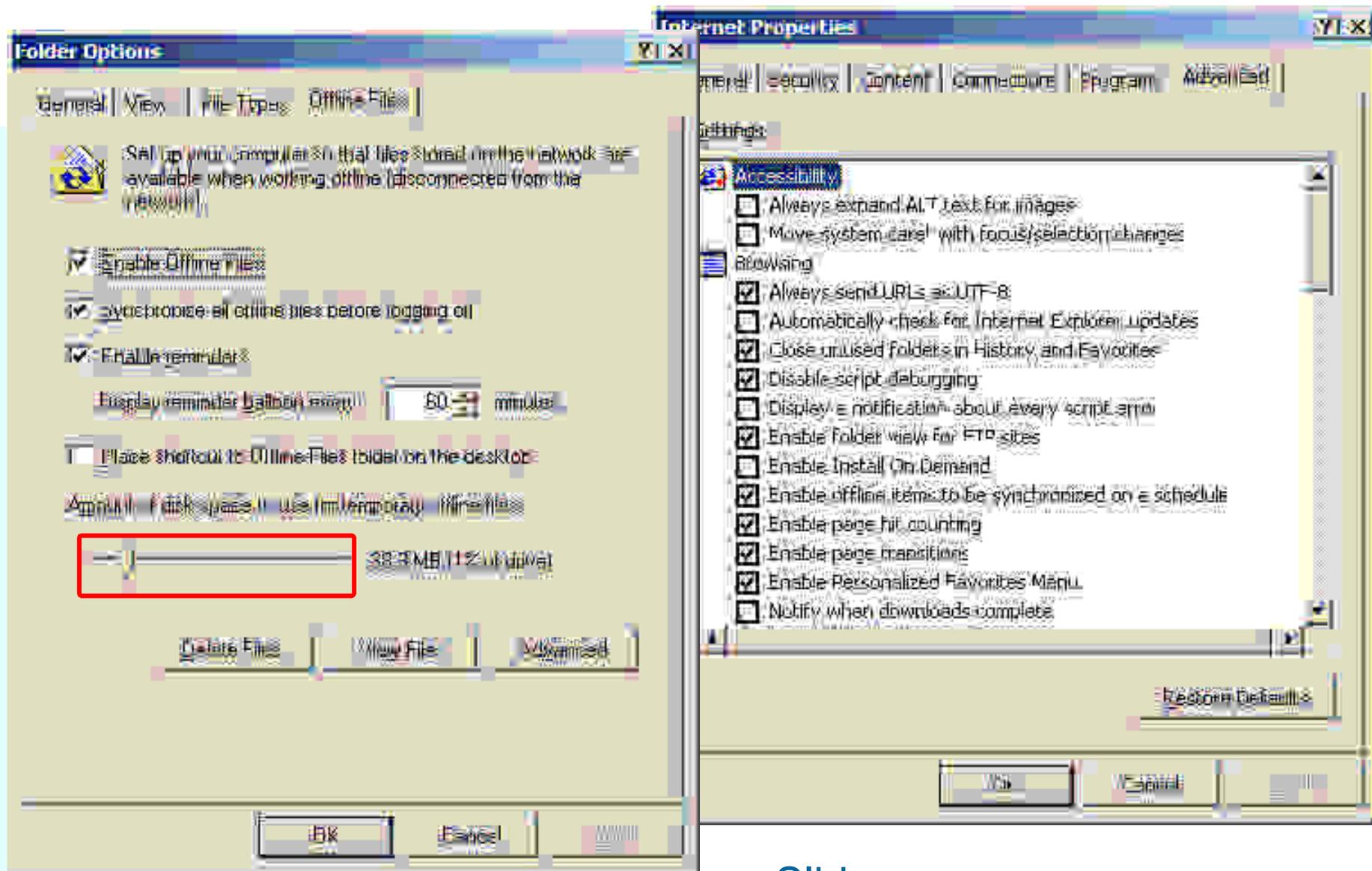
Controlli: altro (1)



Tab



Controlli: altro (2)

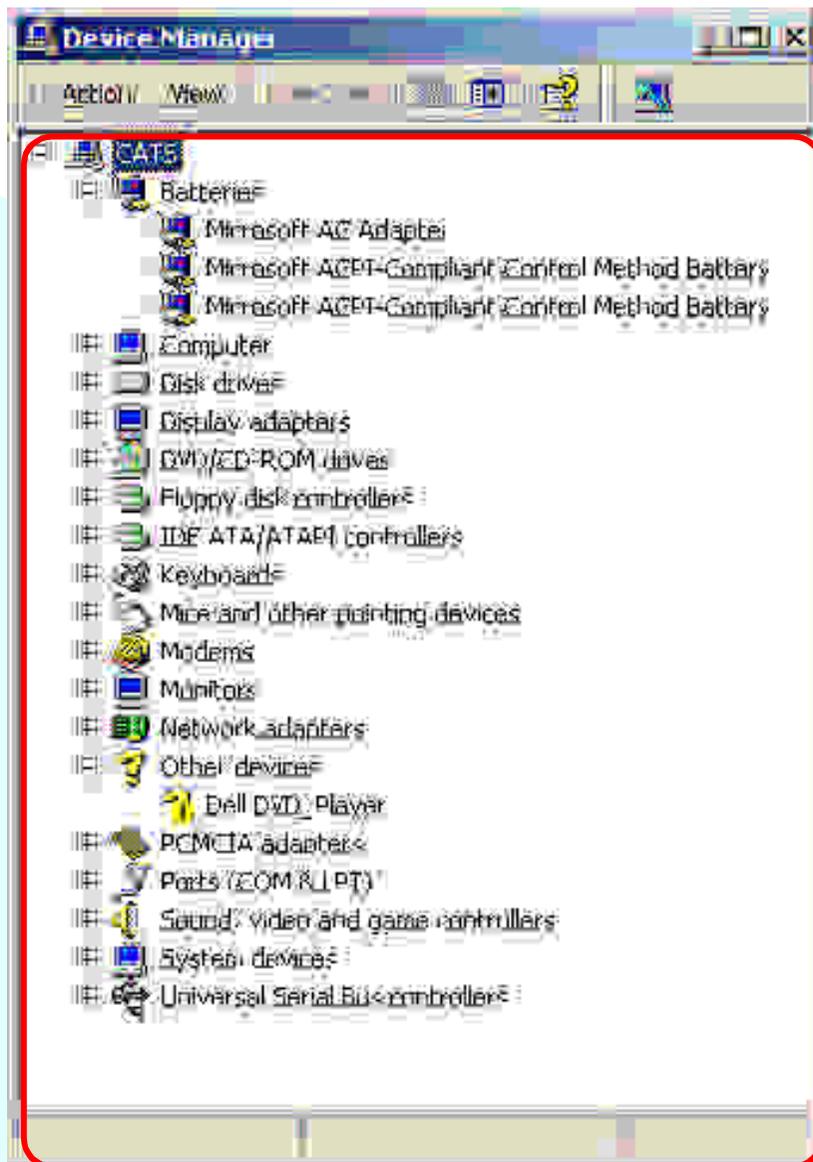


Slider



Controlli: altro (3)

Tree view



Well



Altri aspetti

Finestre di dialogo

- ◆ Spostabili o fisse
- ◆ Ingrandibili o no
- ◆ Modali o non-modali
- ◆ Dialoghi: messaggi informativi, avvertimenti, messaggi critici

L'interfaccia a penna

- ◆ Gest standard



X Windows

- L'ambiente X
- Scopi del X Windows System
- Usare X
- Programmare X
- I concetti base di programmazione di X



L'ambiente X (schema)

X Window System è un ambiente software per workstation (su Unix). Offre un insieme ricco e complesso di funzionalità per il programmatore e l'utente di applicazioni.



L'ambiente X (spiegazione)

L'esistenza del protocollo di rete assicura indipendenza dalla rete, dal device e dalla marca del computer

Non ci sono altre interfacce al base window system al di fuori del protocollo di rete.

In generale si usa una libreria a basso livello per l'interfaccia a X. Questa è XLib.

Il dogma di X è che il base window system fornisce meccanismi, non politiche. Es: non impone una scelta tra overlapping o tiling windows



Scopi di X Window System

- Trasparenza di rete
- Indipendenza da marca e modelli
- Capacità di output
- Capacità di input
- Capacità di condivisione di informazioni e di concorrenza



Trasparenza di rete

Le applicazioni X che girano su una cpu possono mostrare l'output a un display connesso alla stessa cpu, o a qualunque altra.

L'applicazione gira sulla cpu più conveniente, come un data server per applicazioni data-intensive, o un super-computer per applicazioni computation-intensive

Il programma è identico in ogni caso



Indipendenza da marca e modelli

Le applicazioni X sono portabili. Esse interagiscono solo con X, non con l'hardware sottostante.

Non è mai necessario ricompilare o rilinkare un'applicazione quando viene cambiato il device di output.

E' necessario ricompilarla (ovviamente) quando cambia l'hardware o il sistema operativo in cui il programma
GIRA



Capacità di output

Gli schermi sono una gerarchia di finestre sovrapposte, che si possono muovere, spostare, impilare a piacere.

Capacità di disegno grafico: le operazioni sono immediate, non a cache: la workstation non memorizza liste di operazioni grafiche da svolgere tutte insieme, ma le effettua immediatamente. Le operazioni grafiche sono bitmap-oriented.

- ◆ X disegna testo di alta qualità
- ◆ X gestisce sia colori che bianco-e-nero
- ◆ X gestisce immagini grafiche, e può visualizzarle e manipolarle.



Capacità di input

Gli utenti generano input battendo tasti sulla tastiera e manipolando il mouse della workstation. Gli input sono visti come eventi per le applicazioni, che X pone in una coda cosicché l'applicazione possa considerarli in ordine ed agire di conseguenza.

Alcuni eventi modificano la gerarchia delle finestre. Le applicazioni debbono agire di conseguenza

Gli input del mouse vengono mandati all'applicazione corretta direttamente da X, anche quando molte applicazioni stanno funzionando allo stesso tempo

X supporta direttamente vari tipi di tastiere e di mouse



Capacità di condivisione di informazioni e di concorrenza

Le applicazioni possono usare cut e paste per scambiarsi semplici tipi di dati (es. testo)

Per tipi di dati più complessi è possibile che applicazioni diverse condividano dati esplicitamente attraverso le proprietà X.

Le selezioni sono un modo sofisticato e complesso di interazione tra le applicazioni

Varie applicazioni possono essere attive allo stesso tempo sullo schermo. In generale (ma non necessariamente) ogni applicazione avrà una o più finestre proprie. Ogni applicazione opera indipendentemente dalle altre.



Usare X

L'utente è in controllo di una workstation con un mouse, una tastiera, ed uno o più schermi grafici. Ogni finestra mostra l'output di una applicazione (chiamata client) che gira sulla stessa cpu o su un'altra.

la gestione delle finestre è garantita da una applicazione specifica, il window manager, che stabilisce l'aspetto esteriore delle finestre (le politiche specifiche di interfaccia uomo-macchina) e la gestione delle finestre.

Esistono vari window manager per X. Un window manager è un'applicazione come tutte le altre che gira nello spazio dell'utente. Viene attivata da riga comandi o nello script di login alla workstation.



Programmare X

X è scritto in C

La libreria Xlib permette ad applicazioni C di utilizzare X, in particolare per:

- ◆ Ricevere eventi dalla workstation
- ◆ Reagire appropriatamente a questi eventi

Il main event loop è fatto in modo da accettare un evento dalla workstation, decodificarlo e reagire di conseguenza.

Protocollo X: la definizione formale del X Window System è data in termini di un protocollo di comunicazione tra l'applicazione e la workstation.

Toolkits: sono librerie di subroutine di interfaccia uomo-macchina. Esso permette di standardizzare e modularizzare le azioni comuni tra diverse applicazioni. Ogni window manager fornisce un toolkit.

- ◆ Es: X Toolkit (MIT e DEC), Xlib (HP), ATK (Andrew Toolkit, Carnegie-Mellon e IBM), o più recentemente GDK+ (e GNOME)



Concetti di X

Il protocollo di rete X
L'interfaccia procedurale a XLib
La connessione al display
Le risorse
Caratteristiche comuni alle risorse
Gli eventi
Gli errori



Il protocollo di rete X

Viene usato per lo scambio di istruzioni tra applicazione e display. Definisce le caratteristiche di X. Non dipende dal tipo di rete sottostante

Vi sono quattro tipi di messaggi:

- ◆ One way protocol request messages
- ◆ Round-trip protocol request messages
- ◆ Event messages
- ◆ Error event messages



One way protocol request messages

La maggior parte dei messaggi sono a senso unico: non richiedono una risposta positiva o negativa da parte del display.

- ◆ Es.: XDrawLine

L'applicazione manda uno o più messaggi di questo tipo ed è immediatamente libera di fare altre cose.

Le informazioni one-way vengono bufferizzate e spedite in batch. Il tutto è trasparente e mantiene l'ordine dei messaggi.

Si può forzare lo scarico del buffer con il comando di flush.



Round-trip protocol request messages

Alcuni comandi richiedono risposta dal display.

- ◆ Es.: XQueryPointer

In questo caso XLib blocca l'applicazione fino al ricevimento della risposta. Non c'è buffer, ma i comandi vengono immediatamente spediti.

E' necessario prima fare il flush dei comandi a senso unico precedenti.



Event messages

Gli eventi sono cose che succedono all'applicazione e sono al di fuori del controllo dell'applicazione.

- ◆ Es. Azioni dell'utente.

Ogni applicazione sollecita eventi di uno o più tipi

Il display manda tutti e soli gli eventi del tipo richiesto

- ◆ Es. doppio click, tastiera, ma non movimento del mouse, etc.

Poiché il messaggio ha origine al display, e non all'applicazione, non è un messaggio di tipo one-way, ma un messaggio di tipo evento. In più ci sono limiti di dimensioni, e di contesto (un messaggio di evento deve essere self-contained)



Error event messages

Per la natura dei messaggi one-way, può passare del tempo prima che ci si accorga di un errore, e l'applicazione può star facendo cose completamente diverse fidando del fatto che le cose precedenti sono andate a buon fine. Sorge dunque il problema di avvisare l'applicazione dell'esistenza di un errore

- ◆ Es.: Hai disegnato in una finestra non-esistente

Il display genera un evento di errore, simile all'evento di utente, che contiene informazioni sull'errore avvenuto.

Non dipendono solo da errori di programmazione. es: riempimento della memoria della workstation.



L'interfaccia procedurale a XLib

Xlib nasconde i dettagli del protocollo X ai programmatori.

- ◆ Funzioni di richiesta
 - ✦ Interfaccia diretta a funzionalità del protocollo X
- ◆ Funzioni di convenienza
 - ✦ Interfaccia più efficiente a specifiche funzionalità
 - ✦ Es.: XSetForeground pone ad un certo valore il colore di foreground, che può essere anche settato con l'attribuzione alla finestra di un nuovo contesto grafico con XChangeGC
- ◆ Funzioni di servizio
 - ✦ Alcune funzioni non generano richieste di protocollo, ma esaminano le strutture dati locali.
 - ✦ Es.: Lettura della coda di eventi
- ◆ Macro e funzioni informative
 - ✦ Informazioni descrittive sulle caratteristiche del display
 - ✦ Es.: RootWindow o DefaultScreen



La connessione al display

Ogni applicazione deve stabilire una connessione al display X prima di poter procedere a creare output.

Il display è l'insieme di una tastiera, un mouse, ed uno o più schermi

Il nome del display può essere specificato nella command line, o derivare dalla variabile di environment DISPLAY

Ha forma `host:displaynumber.screennumber`



Le risorse

La creazione di interfacce passa attraverso la creazione, l'uso, la modifica e la distruzione di oggetti informatici chiamati risorse

- ◆ Finestre: Aree rettangolari sullo schermo di una workstation. Servono per la visualizzazione dell'output
- ◆ Contesti grafici (GC): Strutture dati per il mantenimento di attributi importanti riguardo le caratteristiche di una resa grafica di un oggetto, come il colore di foreground, di background, etc.
- ◆ Font: Descrizioni di tipi di caratteri per la visualizzazione su schermo.
- ◆ Mappe di colori (color lookup tables): informazioni grafiche sulla disponibilità di colori di un dato schermo
- ◆ Pixmaps: Parti di memoria simili alle finestre, ma non visualizzate. Utili per fare off-line complesse operazioni di disegno, da trasferire sulla finestra solo ad operazioni concluse.
- ◆ Cursori: Descrittori della forma del puntatore del mouse sullo schermo.



Caratteristiche comuni alle risorse

- Le risorse risiedono sul server X (cioè sulla workstation)
- Le risorse vengono identificate con un resource identifier
- Le risorse sono condivisibili tra tutte le applicazioni di una workstation
- Le risorse sono create tramite richieste a senso unico
- Le risorse sono effimere: vengono distrutte all'uscita dell'applicazione
- Le risorse sono a buon mercato: se ne possono usare a dozzine.



Gli eventi

Ogni azione dell'utente (la mossa del mouse, la pressione del tasto della tastiera, etc.) genera un evento, che viene memorizzato, con un time stamp e i parametri che lo caratterizzano, in una coda di eventi chiamata event queue.

Un applicazione non riceve mai un evento che non abbia richiesto.

Due eccezioni: MappingNotify (se si cambia la disposizione dei tasti o del numero di tasti del mouse) ed eventi tra applicazioni.

Gli eventi provengono dalla cosiddetta finestra di focus, che è precisata in termini non ambigui.

Si può richiedere che gli eventi vengano destinati ad una finestra che non sia la finestra in cui gli eventi avvengono (es. menù)

Ci sono 38 tipi di eventi

- ◆ Eventi di mouse, di tastiera, di finestra, di comunicazione interprocesso, di notifica della finestra, di redirezione degli eventi.



Gli errori

La maggior parte degli errori identificano un errore nella specifica di una risorsa X (Bad Window, Bad Pixmap, BadCursor, BadFont, BadColor, BadGC, etc)

Alcuni errori (BadValue, BadMatch, BadAccess) identificano un uso errato di librerie XLib (parametri sbagliati)

Altri (BadRequest, Badlength, BadImplementation) rivelano problemi del sistema X come è stato implementato nel caso specifico nella libreria X, nel protocollo di comunicazione, o nel server della workstation

Uno solo, BadAlloc, specifica che alcune risorse (tipicamente la memoria) sono state completamente consumate.



Conclusioni

Oggi abbiamo parlato di

- ◆ Linee guida di interfaccia Windows
- ◆ Una introduzione a X Windows



Riferimenti

- *The Windows Interface Guidelines for Software Design*, Microsoft Press, 1995
- O. Jones, *Introduction to the X Window System*, Prentice Hall, 1989

