

# ToolManager

---

Un'estensione per il Workbench di Amiga

Versione 2.1  
16 Maggio 1993

---

Copyright © 1990-93 Stefan Becker

È permessa la riproduzione e la distribuzione di copie inalterate di questo manuale purchè l'avviso di copyright e questo avviso di autorizzazione siano riportati su tutte le copie.

Non è data alcuna garanzia di nessun tipo che i programmi descritti in questo documento sono funzionanti al 100%. Utilizzate questo materiale a vostro rischio. L'autore **non** è responsabile per qualsiasi danno che può essere causato dall'utilizzo di questi programmi.

Questo package è liberamente distribuibile, ma è comunque coperto da copyright da Stefan Becker. Questo significa che potete copiarlo liberamente finchè non richiedete più di quanto dovuto per la copia. Questo contributo **non può** essere maggiore di 5 dollari USA o 5 marchi tedeschi.

**Questa limitazione si applica anche ai rivenditori di Public-Domain tedeschi!!**

È permesso includere questo package nelle collezioni Public-Domain, in particolar modo nella libreria Amiga Disk di Fred Fish (incluse le versioni CD ROM della stessa). Il file di distribuzione può essere trasmesso a sistemi di messaggeria (BBS) o server FTP. Se volete ridistribuire questo programma **dovete** utilizzare gli archivi di distribuzione originali 'ToolManager2\_1bin.lha', 'ToolManager2\_1gfx.lha' e 'ToolManager2\_1src.lha'.

Nè i programmi nè i codici sorgenti (nè parte di essi) possono essere inclusi o utilizzati in programmi commerciali senza il permesso scritto dell'autore.

Nè i programmi **nè** i codici sorgenti (nè parte di essi) possono essere utilizzati su qualsiasi macchina utilizzata per la ricerca, lo sviluppo, la costruzione, il collaudo o la produzione di armi o di altre applicazioni militari. Questa limitazione include anche qualsiasi macchina utilizzata per l'addestramento di personale impiegato in uno **qualsiasi** degli scopi sopra esposti.

# 1 Note importanti

Benvenuti nel magnifico mondo di ToolManager 2.1 :-)

- ToolManager e i suoi concetti sono stati modificati drasticamente (vedi Appendice B [Storia], pagina 33) dalla release 1.5.
- A partire dalla release 2.0 di ToolManager, questo programma ha un'opzione *GiftWare*. Se vi piace il programma e lo usate molto spesso dovreste considerare l'invio di una piccola donazione per compensare il lavoro svolto dall'autore nel realizzare questo programma. Vi suggerisco una donazione di 10-15 dollari USA oppure di 10-20 marchi tedeschi. Per cortesia non inviate assegni od ordini di pagamento al di fuori dell'Europa poichè molto spesso l'incasso di questi mezzi di pagamento costa più dell'importo stesso.

Se non inviate una donazione o non siete in grado di farlo, non vi dovete sentire in colpa per questo. In ogni caso, inviatemi un messaggio dicendo che state utilizzando ToolManager (mi piace ricevere posta divertente :-). Vedi Capitolo 2 [Indirizzi], pagina 2.

- Gli utilizzatori di ToolManager 1.X/2.0 possono iniziare con il capitolo di installazione rapida (vedi Capitolo 3 [Installazione rapida], pagina 3. Alcune delle caratteristiche non sono cambiate e il resto è facilmente intuibile semplicemente provando e sbagliando. Per una descrizione dettagliata dei nuovi concetti e caratteristiche leggete la sezione di riferimento di questo manuale (vedi Capitolo 8 [Oggetti], pagina 13).

**Dovete** interrompere qualsiasi processo di ToolManager 1.X/2.0 attivo o la nuova versione non funzionerà. La nuova versione non è in grado di leggere il formato dei file di configurazione della versione 1.X (spiacente).

- I nuovi utenti dovrebbero leggere l'intero documento per capire i concetti e lo scopo del programma. Iniziate con l'Capitolo 4 [Introduzione], pagina 4.
- ToolManager 2.1 utilizza alcune caratteristiche di AmigaOS Release V38 (e successive) e supporta le nuove caratteristiche di rete di AmigaOS, che saranno (si spera) disponibili presto a tutta l'utenza Amiga. Se state ancora utilizzando la Release 2.0 (indicata come V37 in questo documento), non dovete preoccuparvi in quanto ToolManager non si basa su queste caratteristiche. Tutte le caratteristiche estese sono evidenziate in questa documentazione.

## 2 Dove mandare bug reports, commenti e donazioni

L'autore può essere raggiunto al seguente indirizzo:

Indirizzi postali:

Stefan Becker  
Holsteinstrasse 9  
5100 Aachen  
GERMANY

Per cortesia utilizzate il seguente indirizzo dopo il 1° luglio 1993:

Stefan Becker  
Holsteinstrasse 9  
52068 Aachen  
GERMANY

InterNet Electronic Mail:

`stefanb@pool.informatik.rwth-aachen.de`

## 3 Come installare rapidamente ToolManager 2.1

L'installazione di base di ToolManager 2.1 consiste nelle seguenti quattro parti:

`'Libs/ToolManager.library' ⇒ 'LIBS:'`

Questo è il programma principale di ToolManager. Gestisce tutti i programmi, i menu, le icone e i dock (vedi Capitolo 10 [Libreria], pagina 25).

`'Prefs/ToolManager*' ⇒ 'SYS:Prefs'`

Questo è l'editor per la configurazione (vedi Capitolo 9 [Preferenze], pagina 21).

`'WBStartup/ToolManager*' ⇒ 'SYS:WBStartup'`

Con questa utility potete avviare e interrompere ToolManager. Se la mettete nel cassetto WBStartup, ToolManager verrà sempre lanciato quando la vostra macchina viene riavviata.

`'L/WBStart-Handler' ⇒ 'L:'`

Questo programma lancia i programmi secondo il metodo di avvio del Workbench. È un processo separato in modo tale che potete interrompere ToolManager anche quando avete lanciato dei programmi con il metodo WB.

Dopo aver copiato questi file, dovrete uscire da qualsiasi versione di ToolManager che stia girando sulla vostra macchina e fare doppio click sull'icona di ToolManager che risiede nel cassetto `'WBStartup'`. Ora potete lanciare l'editor delle preferenze e provare (utilizzate il bottone "Prova" invece del bottone "Usa" durante le prove). Dovreste essere in grado di capire molte delle caratteristiche semplicemente provando e sbagliando, per maggiori informazioni date un'occhiata alla descrizione degli oggetti di ToolManager (vedi Capitolo 8 [Oggetti], pagina 13).

La distribuzione include una configurazione di esempio chiamata `'TM_Demo.prefs'`. Potete caricarla nell'editor di preferenze con la voce di menu **Carica**.

## 4 Che cosa è ToolManager?

ToolManager è un flessibile programma che permette di gestire gli strumenti (tool) nel vostro ambiente di lavoro. Può lanciare programmi Workbench e CLI, script ARexx e generare eventi Hot Key. Può anche inviare comandi ad un processo ToolManager attivato su una macchina remota. L'interfaccia utente è composta da menu, icone e finestre dock. Se vi piace un computer rumoroso, potete associare un suono a ciascuno di questi elementi. Vedi Paragrafo 8.3 [Suono], pagina 15.

ToolManager può aggiungere voci al menu **Strumenti** del Workbench. Se selezionate una di queste voci, verrà lanciato il programma associato. Ogni icona precedentemente selezionata verrà utilizzata come un argomento per il programma. Questa caratteristica è solamente disponibile quando il Workbench è attivo. Vedi Paragrafo 8.4 [Menu], pagina 15.

ToolManager è in grado di aggiungere icone alla finestra del Workbench. Quando fate doppio click su una di queste icone, il programma associato verrà lanciato. Se lasciate cadere altre icone su una di queste icone, il programma verrà avviato avente come argomenti le icone lasciate cadere. Questa caratteristica è solamente disponibile quando il Workbench è attivo. Vedi Paragrafo 8.5 [Icona], pagina 16.

ToolManager può creare una finestra dock da una collezione di programmi. Questa finestra può essere aperta su qualsiasi schermo pubblico. Ogni programma è rappresentato da un'immagine o da un bottone. Per lanciare il programma basta semplicemente fare click sull'immagine o sul bottone. Se la finestra dock è stata aperta sullo schermo del Workbench e il Workbench è attivo, potete anche lasciare cadere alcune icone sull'immagine o sul bottone per attivare il corrispondente programma con degli argomenti. Vedi Paragrafo 8.6 [Dock], pagina 17.

Inoltre potete assegnare un Hot Key ad ogni programma. Se premete questo tasto, il programma verrà lanciato. Notate che *non* può essere passato alcun argomento al programma se utilizzate questo metodo di attivazione. Vedi Paragrafo 8.1 [Programma], pagina 13.

## 5 I concetti alla base di ToolManager

ToolManager 2.1 utilizza un nuovo tipo di approccio orientato agli oggetti (object oriented) per fornire un sistema flessibile ed estensibile. Questo approccio ha reso possibile il miglioramento di diverse caratteristiche presenti nelle versioni 1.X di ToolManager; per esempio, potete ora avere contemporaneamente più finestre dock.

Un oggetto è una collezione di dati che descrivono le sue caratteristiche. Ogni oggetto ha un nome ed un tipo. Potete creare un qualsiasi numero di oggetti di qualsiasi tipo, ma il nome di ogni oggetto deve essere unico poichè è utilizzato come riferimento ad un determinato oggetto.

Attualmente ci sono sette tipi differenti di oggetto: Programma, Immagine, Suono, Menu, Icona, Dock e Accesso. I primi tre sono oggetti di base; questo significa che non fanno riferimenti ad altri oggetti. Essi forniscono dati o servizi per gli oggetti di tipo complesso.

Gli ultimi quattro oggetti sono di tipo complesso; essi fanno riferimento agli oggetti di base e utilizzano questi ultimi per accedere a dati o servizi. Il riferimento è fatto per nome, e se non esiste alcun oggetto di base con quel nome, l'oggetto complesso lo ignorerà. Notate che questo può ridurre la funzionalità degli oggetti di tipo complesso: per esempio, un oggetto Icona *necessita* dei dati da un oggetto Immagine, sicchè se quest'ultimo non esiste l'oggetto complesso non creerà un'icona.

Per una descrizione dettagliata di tutti i parametri degli oggetti fate riferimento a Capitolo 8 [Oggetti], pagina 13.

## 6 Un tour guidato attraverso ToolManager

Non avete capito una sola parola fino adesso? Siete confusi da oggetti, programmi e riferimenti? Non vi scoraggiate, eccovi un aiuto.

Vi guiderò ora attraverso un esempio eseguito passo passo che dimostrerà come configurare ToolManager. Tutto quello di cui avete bisogno è installare ToolManager e lanciare l'editor delle preferenze. Dopo ogni passo, utilizzate il bottone “Prova” nella finestra principale per verificare la configurazione.

Come esempio utilizzeremo il programma di visualizzazione testi More che risiede nel cassetto ‘SYS:Utilities’. Per prima cosa dobbiamo dire a ToolManager quale programma intendiamo utilizzare. Le informazioni sui programmi sono conservate negli oggetti Programma. Selezionate per cui “Programma” come tipo di oggetto nella finestra principale dell'editor delle preferenze e premete il bottone “Nuovo”.

Dopo aver premuto il bottone vedrete apparire la finestra “Modifica programma”. Aprite ora il cassetto Utilities nella vostra partizione del Workbench, muovete l'icona di More fuori dal cassetto e lasciatela cadere nella finestra dell'editor. Come potete vedere, l'editor ha ora impostato il nome dell'oggetto e il comando con il nome del programma, e la directory corrente a System:Utilities (NdT se System è il nome della vostra partizione). Premete il bottone “OK” per accettare le impostazioni effettuate.

Non potete fare molto con il solo oggetto Programma, per cui come prossimo passo aggiungeremo questo programma al menu “Strumenti” del Workbench. Selezionate “Menu” come tipo di oggetto e premete il bottone “Nuovo”. Dovreste ora vedere la finestra “Modifica menu”. Cambiate il nome dell'oggetto in “Visualizza testo”.

ToolManager ha bisogno di sapere quale programma deve lanciare quando la voce di menu è selezionata, per cui noi ora agganceremo un oggetto Programma all'oggetto Menu. Premete il bottone “Programma...” e selezionate l'oggetto “More” dal requester. Premete ora il bottone “OK” e quindi il bottone “Prova” nella finestra principale. Potete ora verificare che nel menu “Strumenti” del Workbench c'è una nuova voce. Selezionate un file di testo tramite il Workbench e scegliete la nuova voce del menu. Il programma “More” dovrebbe ora essere lanciato e visualizzare il file di testo preselezionato. Facile, vero?

Possiamo ora fare un ulteriore passo e creare un oggetto Icona sul Workbench. Per visualizzare un'icona abbiamo bisogno dei dati di un'immagine, che sono memorizzati in un oggetto Immagine. Selezionate “Immagine” come tipo di oggetto e premete il bottone “Nuovo”. La finestra “Modifica



immagine” si aprirà. Cambiate il nome in “Immagine per More” e lasciate cadere l'icona di More dal cassetto Utilities nella finestra dell'editor. Premete “OK” per utilizzare queste impostazioni.

Nel prossimo passo creeremo un oggetto Icona. Selezionate “Icona” come tipo di oggetto e premete il bottone “Nuovo”. Cambiate il nome dell'oggetto in “Mostra testo”. Premete il bottone “Programma...” e selezionate l'oggetto “More” dal requester. Premete quindi il bottone “Immagine” e selezionate l'oggetto “Immagine per More” dal requester. Impostate la posizione X a 100 e la posizione Y a 50. Premete il bottone “OK” e quindi il bottone “Prova”. Dopo un piccolo ritardo un'icona apparirà sul Workbench, sulla quale potete lasciar cadere le icone dei vostri file di testo in modo da visualizzarli.

Sono sicuro che ora avete un'idea di come utilizzare gli oggetti di ToolManager e in quale modo dovete unirli insieme per costruire il vostro ambiente di lavoro. Ora potete scoprire le rimanenti caratteristiche provandole una a una. Potete anche vedere la configurazione dimostrativa contenuta nel file ‘TM\_Demo.prefs’.

## 7 Descrizione di tutti i file nella distribuzione

La distribuzione completa di ToolManager 2.1 consiste di diverse directory che vengono spiegate qui sotto. Notate che la distribuzione è suddivisa in tre parti, sicchè vi potrebbe capitare di non avere tutte le directory che vengono ora menzionate.

### 7.1 La directory Docs

Questa directory contiene la documentazione relativa a ToolManager. La documentazione è disponibile in quattro differenti formati e in diverse lingue. È incluso inoltre un file in formato AutoDoc che descrive l'interfaccia della libreria shared di ToolManager.

Prefix `'TM_<lingua>'`

Questo file contiene la documentazione per una determinata lingua. Attualmente sono disponibili le lingue Deutsch, English, Français, Italiano e Svenska.

Postfix `' .doc'`

Questo file contiene la documentazione in formato ASCII puro.

Postfix `' .dvi'`

Questo file contiene la documentazione in formato T<sub>E</sub>X DVI. Per ottenere un manuale stampato elaborate questo file con un driver di stampa di T<sub>E</sub>X.

Postfix `' .guide'`

Questo file contiene la documentazione in formato AmigaGuide. Sebbene esso consista solamente in un testo ASCII con alcuni comandi, avete bisogno di AmigaGuide per utilizzare i riferimenti iper-testo in esso contenuti.

Postfix `' .tex'`

Questo file contiene la documentazione in formato Texinfo, secondo le specifiche della Free Software Foundation (FSF). Unitamente al package di macro `'texinfo.tex'`, potete utilizzare T<sub>E</sub>X e `'texindex'` per creare un file in formato DVI (vedi sopra).

`'ToolManager.doc'`

Questo file contiene, in formato AutoDoc, la descrizione dell'interfaccia della libreria di ToolManager.

## 7.2 La directory Goodies

Questa directory contiene package di programmi addizionali che sono utili per l'operatività di ToolManager.

`'GetPubName.lha'`

Questo piccolo programma scrive il nome dello schermo pubblico in primo piano nel flusso standard di output (stdout) o in una variabile di environment. Scritto da Michael "Mick" Hohmann.

`'upd1_20.lha'`

Il programma `upd` è stato scritto da Jonas Petersson. È un piccolo programma che apre una porta ARexx e attende dei comandi. Tramite ARexx potete ordinare a `upd` di suonare file contenenti suoni campionati. ToolManager utilizza questa caratteristica per implementare i suoi oggetti Suono. Vedi Paragrafo 8.3 [Suono], pagina 15.

## 7.3 La directory Graphics

Questa directory contiene una ricca collezione di immagini tra le quali potete scegliere le vostre favorite. Caricatele semplicemente come oggetti Immagine in ToolManager (vedi Paragrafo 8.2 [Immagine], pagina 14).

I file sono stati forniti come contributo da diverse persone (vedi Appendice C [Ringraziamenti], pagina 34). Ognuno di essi è contenuto in una separata sotto-directory dell'albero di distribuzione. Dal momento che sono stati creati da vari autori i file hanno caratteristiche differenti (palette, numero di bitplane, risoluzione, dimensioni) e hanno differenti stili di disegno. Quindi non tutte le immagini potranno apparire gradevoli sulla vostra macchina.

Per differenziare i formati di immagine che sono supportati da ToolManager, ogni file ha un'estensione che descrive il formato dei file:

- `' .anmb'` Questo è un file IFF ANIM creato da un programma di disegno/animazione e può contenere diverse figure. Sebbene ToolManager sia in grado di caricare file ANIM completi, dovete utilizzare qualcosa tipo la caratteristica "AnimBrush" di Deluxe Paint per ritagliare le parti interessanti dell'animazione.
- `' .brush'` Questo è un file IFF ANIM creato da un programma di disegno. Contiene solamente un'immagine.

‘.info’      Questa è una normale icona di Amiga creata con IconEdit (o qualcosa di simile). Può contenere due immagini.

## 7.4 La directory L

Questa directory contiene solo un file, ‘WBStart-Handler’. Dovete copiare questo file nella directory ‘L:’, in caso contrario ToolManager non sarà in grado di lanciare alcun oggetto Programma utilizzando il metodo di lancio WB (vedi Paragrafo 8.1 [Programma], pagina 13).

Il package completo WBStart 1.2 può essere trovato sul Fish Disk #757.

## 7.5 La directory Libs

Questa directory contiene solamente un file, ‘ToolManager.library’. Questo è il programma principale di ToolManager e deve essere copiato nella directory ‘LIBS:’.

## 7.6 La directory Locale

Questa directory contiene tutti i file per la localizzazione di ToolManager. Dal momento che la locale.library è distribuita per la prima volta con la V38, non dovete copiare questi file se state utilizzando la V37. Se state utilizzando la V38, scegliete solamente i file relativi alla vostra lingua e copiateli nelle directory appropriate.

‘Catalogs/<lingua>/ToolManager.catalog’

Questo è un file di traduzione per la lingua specifica. Copiate il file per la vostra lingua nella directory ‘LOCALE:Catalogs/<lingua>’.

‘Languages/<lingua>.language’

Alcune lingue non sono supportate nella distribuzione standard della V38, così alcune traduzioni sono state fornite sotto forma di file ‘.language’, in modo che ToolManager possa utilizzare anche queste lingue. Copiate il file per la vostra lingua nella directory ‘LOCALE:Languages’. Le lingue aggiuntionali disponibili sono: Finnish (suomi), Eefeler Platt (eifel).

## 7.7 La directory Prefs

L'editor di preferenze di ToolManager e la sua icona risiedono in questa directory. Copiate entrambi i file nella directory `'SYS:Prefs'`. Per maggiori informazioni sull'editor fate riferimento a Capitolo 9 [Preferenze], pagina 21.

## 7.8 La directory Programmers

Questa directory contiene tutti i file che sono utilizzati dai vari linguaggi e i loro compilatori per accedere all'interfaccia della libreria shared di ToolManager. Guardate nella sotto-directory `'examples'` per alcuni esempi sul come usare questa interfaccia. Per una completa descrizione dell'interfaccia leggete il file `'Docs/ToolManager.doc'`.

Attualmente i linguaggi/compilatori supportati sono: AmigaOberon, DICE C, M2Amiga Modula-2, MANX Aztec C and SAS C.

## 7.9 La directory Scripts

Questa directory contiene una collezione di script per ARexx o la Shell che possono essere utilizzati come oggetti Programma di ToolManager. Notate che questi script possono essere specifici per la shell o l'ambiente di lavoro degli autori originari, per cui potrebbe rendersi necessaria qualche modifica.

## 7.10 La directory Source

Questa directory contiene il codice sorgente completo di ToolManager 2.1 e le sue utility. Ogni programma ha la sua sotto-directory. L'autore fornisce il codice sorgente come un esempio di programmazione per i sistemi operativi 2.x/3.0.

La sotto-directory `'locale'` è interessante solo per i traduttori. Se la vostra lingua non è supportata in questa versione e volete fare una traduzione delle stringhe utilizzate da ToolManager guardate il file `'empty.ct'`. Riempite le linee vuote e mandatemi il file risultante. Probabilmente verrà incluso nella prossima versione.

## 7.11 La directory WBStartup

Solamente un programma è contenuto in questa directory: **ToolManager**. Questa utility inizializza e termina ToolManager 2.1. Il più delle volte questa utility risiede nella directory **'SYS:WBStartup'** (e quindi viene attivata automaticamente ad ogni reset della macchina), ma può essere utilizzata anche dalla Shell.

## 8 Manuale di riferimento degli oggetti di ToolManager

Questo capitolo descrive in dettaglio gli oggetti di ToolManager. Ogni oggetto ha un tipo e un nome: quest'ultimo è utilizzato per identificarlo. Ci sono sette tipi differenti di oggetti:

### 8.1 Oggetti programma

Gli oggetti Programma descrivono i programmi o le azioni che devono essere avviate da ToolManager. Sono supportati tre tipi differenti di programmi: CLI, Workbench e ARexx. Sono supportate tre tipi differenti di azioni: Dock, Hot Key e Network (Rete). Ogni oggetto Programma offre i seguenti parametri (i valori di default sono riportati tra parentesi):

#### Tipo programma (CLI)

Questo parametro specifica il tipo di programma o azione. Può essere uno fra i seguenti: CLI, WB (Workbench), ARexx, Dock, Hot Key o Network (Rete).

**Comando** Il nome del file del programma o dell'azione da lanciare. Questo nome può essere relativo alla directory corrente. Se il tipo del programma è Dock il comando descrive il nome dell'oggetto dock che deve essere attivato o disattivato. Per il tipo programma Hot Key il nome del comando deve essere una stringa di descrizione di input secondo le specifiche delle Commodities Exchange (vedi Capitolo 11 [Hot Key], pagina 26). Un comando remoto (tipo programma Network) è descritto come `NomeOggetto@NomeMacchina` e indica ad un processo ToolManager attivo sul computer `NomeMacchina` di attivare l'oggetto Programma chiamato `NomeOggetto`.

**Hot Key** Potete impostare un Hot Key per ogni oggetto Programma. Se viene generato un evento di input corrispondente a questo parametro, il programma verrà lanciato. Nota: il programma verrà attivato senza argomenti.

#### Stack (4096)

Questo parametro imposta la dimensione dello stack iniziale del programma.

#### Priorità (0)

Questo parametro imposta la priorità di esecuzione iniziale del programma.

#### Ritardo (0)

Dopo l'attivazione di un oggetto Programma, ToolManager attende un numero di secondi pari al valore di questo parametro prima di lanciare il programma. Se questo valore è negativo, il programma verrà lanciato ogni `Ritardo` secondi. Per interrompere un oggetto Programma che è in attesa di essere eseguito basta attivarlo una seconda volta. Nota: se viene specificato un valore diverso da 0, il programma verrà lanciato senza argomenti.

**Directory corrente ('SYS:')**

Il nome della directory corrente per il programma. Nota: i programmi ARexx ignorano questo parametro.

**Percorso** (Percorso ereditato dal processo di ToolManager)

Questa stringa imposta il percorso di ricerca utilizzato per il programma. Potete specificare diverse directory separandone i nomi con un “;”. Questo è utile solamente per i programmi CLI.

**File di output ('NIL:')**

Questo è il nome del file su cui ridirigere l'output. È utile solamente per i programmi CLI.

**Schermo pubblico** (Schermo pubblico di default)

Potete impostare il nome dello schermo pubblico che deve essere portato in primo piano prima che avvenga l'esecuzione del programma. Questo parametro è valido solo in congiunzione con il parametro **Primo piano**.

**Argomenti** (Sì)

Questo parametro controlla la gestione dei parametri del programma. Se un programma non prevede o non ha bisogno dei parametri, potete disabilitare il passaggio degli argomenti.

**Primo piano** (No)

Se selezionate questo parametro lo schermo pubblico specificato nel parametro **Schermo pubblico** verrà portato in primo piano prima che avvenga l'esecuzione del programma.

## 8.2 Oggetti immagine

Gli oggetti immagine specificano i dati dell'immagine che devono essere utilizzati da ToolManager per le icone o per i dock. Questo tipo di oggetto ha un solo parametro:

**Nome file** Questo parametro specifica il nome del file dal quale ToolManager deve leggere i dati dell'immagine. ToolManager prova a riconoscere il tipo dei dati dell'immagine automaticamente.

1. Prova a caricare il file come se fossero dati IFF. Attualmente ToolManager è in grado di leggere file ILBM (una immagine) o ANIM (due o più immagini).
2. Prova a leggere il file come un'icona. Un'icona può avere una o due immagini.



Le animazioni sono attualmente supportate solo dagli oggetti Dock. Gli oggetti Icona selezionano solo la prima e la seconda immagine dell'animazione per costruire un'icona a due immagini. Se volete creare un'animazione per ToolManager, dovrete seguire le seguenti regole:

Immagine 1

Questa dovrebbe essere un'immagine che rappresenta lo stato inattivo.

Immagine 2

Questa dovrebbe essere un'immagine che rappresenta lo stato selezionato. Normalmente è una copia invertita della prima immagine.

Immagine 3 fino a N-1

Queste sono immagini per l'animazione. Ogni immagine verrà visualizzata per un 1/3 di secondo.

Immagine N

L'ultima immagine dell'animazione verrà mostrata per un secondo, dopodichè verrà ripetuta l'intera sequenza.

## 8.3 Oggetti suono

Un oggetto Suono può essere utilizzato per sonorizzare alcuni aspetti di ToolManager. Dal momento che ToolManager non ha alcuna funzionalità integrata per suonare dati campionati, esso utilizza ARexx per attivare un processo in background in grado di riprodurre suoni. Questo tipo di oggetto ha due parametri:

**Comando**     Questo parametro imposta il comando ARexx che ToolManager deve inviare per attivare il processo esterno di riproduzione suoni. Per l'utility `upd` il valore di questo parametro potrebbe essere qualcosa del tipo `file samples:boing` che comanda a `upd` di suonare il campionamento in formato IFF '`samples:boing`'. Vedi Paragrafo 7.2 [Goodies], pagina 9.

**Porta ARexx**

Questo specifica la porta ARexx a cui ToolManager deve inviare il **Comando**. Il valore di default è `PLAY` che è la porta ARexx dell'utility `upd`.

## 8.4 Oggetti menu

Gli oggetti Menu controllano le voci nel menu "Strumenti" del Workbench. Il nome dell'oggetto è utilizzato per il testo della voce di menu. Per attivare questo tipo di oggetto basta selezionare

la voce nel menu. Gli oggetti Menu funzionano solo quando il Workbench è attivo. Questo tipo di oggetto ha due parametri:

**Programma...**

Questo è il nome di un oggetto Programma che deve essere attivato quando viene selezionata la corrispondente voce del menu. Qualsiasi icona che sia stata selezionata in precedenza verrà utilizzata come un argomento del programma.

**Suono...** Questo è il nome di un oggetto Suono che deve essere attivato quando viene selezionata la voce del menu.

Nota agli utenti di ToolManager 1.X: per simulare il vecchio tooltype “Dummy” basta creare un oggetto Menu e *non* specificare alcun oggetto Programma e Suono.

## 8.5 Oggetti Icona

Gli oggetti Icona descrivono le icone delle applicazioni visualizzate da ToolManager nella finestra del Workbench. Questi oggetti possono essere attivati facendo doppio click sull'icona oppure lasciando cadere altre icone sull'icona stessa. Gli oggetti Icona funzionano solamente quando il Workbench è attivo. I parametri per questo tipo di oggetto sono (i valori di default sono riportati tra parentesi):

**Programma...**

Questo è il nome di un oggetto Programma che deve essere attivato quando viene selezionata l'icona. Ogni altra icona che è stata lasciata cadere sopra l'icona applicativa verrà utilizzata come un argomento del programma.

**Immagine...**

Questo parametro è il nome di un oggetto Immagine. I dati dell'immagine di questo oggetto sono utilizzati per creare l'icona applicativa.

**Suono...** Questo è il nome dell'oggetto Suono che deve essere attivato quando viene selezionata l'icona.

**Bordo sinistro (0)**

Questo valore specifica la posizione sinistra dell'icona applicativa.

**Bordo superiore (0)**

Questo valore specifica la posizione superiore dell'icona applicativa.

**Mostra nome (Yes)**

Se questo parametro è attivato, il nome dell'oggetto verrà utilizzato come nome dell'icona applicativa.

Nota: Il Workbench è *molto* sensibile alla posizione delle icone. Se specificate delle coordinate contrastanti con la disposizione delle icone scelta dal Workbench quest'ultimo le ignorerà e piazzerà l'icona da qualche altra parte.

## 8.6 Oggetti Dock

Gli oggetti Dock descrivono delle finestre. Queste finestre contengono diversi tool (strumenti) ognuno dei quali è rappresentato da immagini o gadget. Per lanciare un determinato strumento basta fare click sulla sua immagine (o gadget). Naturalmente potete lasciar cadere alcune icone sull'immagine per fornire argomenti al programma associato. Ogni oggetto Dock ha diversi parametri (i valori di default sono riportati tra parentesi):

**Titolo finestra**

Questo parametro specifica il titolo della finestra dock. Se fornite un titolo, la finestra dock sarà una normale finestra in stile OS 2.0 con tanto di barra di trascinamento, gadget di chiusura, gadget di profondità e bordo. Se *non* specificate un titolo otterrete una finestra dock composta solamente dalla barra di trascinamento e da *nessun* bordo.

**Hot Key** Potete impostare un Hot Key per ogni oggetto Dock. Se viene generato un evento di input corrispondente al tasto Hot Key lo stato di attivazione della finestra dock verrà invertito, ovvero la finestra verrà chiusa o aperta.

**Bordo sinistro (0)**

Questo parametro imposta la posizione sinistra della finestra dock. Se è impostato anche il parametro **Centrato** allora questo parametro verrà ignorato.

**Bordo superiore (0)**

Questo parametro imposta la posizione superiore della finestra dock. Se è impostato anche il parametro **Centrato** allora questo parametro verrà ignorato.

**Colonne (1)**

Questo parametro imposta il numero di colonne dei tool. Gli strumenti sono sempre ordinati per riga, iniziando dalla colonna all'estrema sinistra e continuando con quelle alla sua destra.

**Schermo pubblico** (Schermo pubblico di default)

Questo parametro specifica lo schermo pubblico sul quale la finestra dock deve essere aperta. Se la finestra è stata aperta tramite un Hot Key lo schermo pubblico specificato verrà spostato in primo piano dopo che la finestra è stata aperta. Questo parametro è ignorato se è stato impostato il parametro **Primo piano**.

**Font...** (Font dello schermo)

Se avete una finestra dock nella quale avete impostato il parametro **Testo** tramite questo parametro potete scegliere il font da utilizzare per i gadget.

**Attivato** (Sì)

Una finestra dock può essere attiva (aperta) o meno (chiusa).

**Sfondo** (No)

Questo parametro indica che la finestra dock deve essere spostata immediatamente dietro a tutte le altre finestre dopo essere stata aperta.

**Centrato** (No)

Se questo parametro è impostato, al momento dell'apertura la finestra dock verrà sempre centrata rispetto alla posizione corrente del puntatore del mouse.

**Primo piano** (No)

Se impostate questo parametro la finestra dock si aprirà sempre sullo schermo pubblico in primo piano.

**Menu** (No) Potete aggiungere un piccolo menu ad ogni finestra dock. Questo menu ha due voci:

**Chiudi Dock** Chiude la finestra dock.

**Fine TM** Termina ToolManager.

**Motivo** (No)

La finestra dock adatta automaticamente la sua dimensione a quella dell'immagine più larga. Ogni elemento della finestra ha la stessa dimensione e le immagini più piccole vengono centrate in modo da avere intorno un bordo vuoto. Se non vi piace il bordo vuoto, impostate questo parametro e il bordo verrà riempito con un motivo.

**Pop Up** (No)

Quando questo parametro è impostato la finestra dock viene chiusa automaticamente non appena viene selezionata un'immagine o un gadget del dock. Questo è utile specialmente in congiunzione con i parametri **Centrato**, **Primo piano** e un Hot Key della classe **rawmouse** (vedi Capitolo 11 [Hot Key], pagina 26).

**Posiz. fissa** (No)

Normalmente una finestra dock memorizza la sua ultima posizione quando la chiudete e riappare nella stessa posizione quando la riaprite. Selezionate questo parametro se volete che la finestra dock sia aperta sempre nella stessa posizione dello schermo.

**Testo (No)**

Con questo parametro potete scegliere se visualizzare delle immagini o dei gadget nella finestra dock. In particolare le finestre dock con i bottoni sono utili quando vengono utilizzate insieme al parametro **Pop Up**.

**Verticale (No)**

Se la finestra dock utilizza il nuovo design (ovvero mostra solamente una barra di trascinamento e nessun bordo), con questo parametro potete impostare l'orientamento della barra di trascinamento. Questo parametro viene ignorato quando specificate un titolo per la finestra dock con il parametro **Titolo finestra**.

## 8.7 Oggetti accesso

Gli oggetti Accesso controllano i diritti di accesso alle richieste che arrivano dalla rete. Per default *qualsiasi* tipo di richiesta viene rifiutata in modo tale che un processo ToolManager remoto non possa interrompere le operazioni della vostra macchina attivando alcuni dei vostri oggetti Programma. Con gli oggetti Accesso potete fare in modo che un ristretto numero di macchine possano attivare i suddetti oggetti Programma.

Il nome di un oggetto Accesso ha un significato speciale, visto che viene confrontato con il nome della macchina remota che ha richiesto un accesso alla vostra macchina. ToolManager usa il seguente schema per eseguire il controllo:

1. Confronta il nome dell'oggetto con quello completo dell'host
2. Confronta il nome dell'oggetto con quello del realm (dominio)
3. Cerca un oggetto Accesso chiamato **anyone**

Se viene trovato un oggetto corrispondente questo viene utilizzato come chiave di accesso per la macchina remota. L'oggetto chiamato **anyone** è utilizzato per una qualsiasi richiesta di rete per la quale non esiste un corrispondente oggetto Accesso.

L'oggetto Accesso ha solamente un parametro:

**Oggetto programma**

Questo parametro può essere utilizzato diverse volte e specifica quali oggetti Programma possono essere attivati da una macchina remota. Se non specificate *alcun*

oggetto allora la macchina remota avrà accesso a *tutti* gli oggetti Programma definiti sulla vostra macchina.

## 9 L'editor di preferenze di ToolManager

Con l'editor di preferenze potete modificare la configurazione di ToolManager. Questa configurazione viene caricata automaticamente ogni volta che attivate ToolManager. Per lanciare l'editor basta fare doppio click sulla sua icona: vedrete quindi apparire la finestra principale dell'editor.

La maggior parte dei gadget nelle finestre dell'editor sono attivabili anche tramite tastiera. I tasti da utilizzare sono evidenziati da una lineetta (\_). Tenete presente che se è attivo un gadget stringa dovete prima premere il tasto di invio (Return) prima di poter utilizzare i suddetti tasti.

### 9.1 Gadget della finestra principale

La finestra principale mostra diversi gruppi di gadget:

#### Tipo oggetto

Con questo gadget ciclico potete selezionare il tipo di oggetto che volete creare o modificare.

#### Lista oggetti

Questo gadget mostra la lista di tutti gli oggetti del tipo richiesto. Potete selezionare un oggetto facendo click sul suo nome. Se fate doppio click su di un elemento della lista si aprirà una finestra per la modifica di quell'oggetto.

#### Spostamento dell'oggetto

Una volta selezionato un oggetto potete muoverlo all'interno della lista con questo gruppo di gadget. Se fate click sul bottone **Riordina** gli elementi della lista verranno riordinati alfabeticamente.

#### Manipolazione dell'oggetto

Questo gruppo di gadget serve per la manipolazione degli oggetti. Il gadget **Nuovo** crea un nuovo oggetto del tipo correntemente selezionato. Facendo click sul bottone **Modifica...** viene aperta una finestra per la modifica dell'oggetto selezionato. Con il gadget **Copia** potete fare una copia dell'oggetto selezionato. Il bottone **Cancella** invece rimuove dalla lista l'oggetto selezionato.

#### Configurazione

Avete diverse possibilità per salvare la configurazione. Con il gadget **Salva** potete salvare la configurazione in modo permanente nel file `'ENVARC:ToolManager.prefs'`. Per modifiche temporanee usate invece il gadget **Usa**, che salva la configurazione del

file `'ENV:ToolManager.prefs'`. Questo file verrà automaticamente cancellato al prossimo reset della macchina. Per provare la nuova configurazione senza lasciare l'editor delle preferenze, usate il gadget **Prova**. Infine, con il bottone **Annulla** potete uscire dall'editor senza salvare le eventuali modifiche.

## 9.2 Menu finestra principale

La finestra principale dell'editor delle preferenze ha diverse voci nei menu:

- |                 |   |
|-----------------|---|
| <b>Progetto</b> | Con le voci <b>Carica...</b> e <b>Salva come...</b> potete caricare e salvare la configurazione. Con la voce <b>Accoda...</b> potete invece caricare un file di configurazione e aggiungere i suoi dati a quelli correntemente attivi. La voce <b>Informazioni...</b> apre un requester contenente le informazioni sul programma, mentre la voce <b>Fine</b> termina l'esecuzione dell'editor senza salvare la configurazione.                  |
| <b>Editor</b>   | Con questo menu potete ripristinare le vecchie configurazioni. La voce <b>Ripristina ultimi valori</b> carica nell'editor la configurazione salvata precedentemente nel file <code>'ENVARC:ToolManager.prefs'</code> . La voce <b>Ripristina valori iniziali</b> permette invece di caricare i dati salvati nel file <code>'ENV:ToolManager.prefs'</code> che sono stati caricati inizialmente nell'editor di preferenze al momento del lancio. |
| <b>Opzioni</b>  | Potete scegliere con la unica voce di questo menu ( <b>Genera icone?</b> se la voce <b>Salva come...</b> del menu <b>Progetti</b> deve creare o meno delle icone.   |

## 9.3 Finestra di creazione oggetti

Se lasciate cadere un'icona sulla finestra principale dell'editor verrà aperta automaticamente la finestra "Crea oggetti". Qui potete scegliere quali tipi di oggetto devono essere creati a partire dal contenuto informativo dell'icona. Questo metodo può essere utilizzato per creare in modo facile e veloce oggetti per i vostri programmi.

Potete, per esempio, creare semplicemente un oggetto Programma e uno Immagine selezionando una delle prime due scelte. Ma potete anche creare un oggetto Menu e/o uno Icona se selezionate una delle ultime tre scelte.



## 9.4 Finestre di modifica

Ogni tipo di oggetto ha una finestra differente per la modifica dei suoi parametri. Per un lista dettagliata di tutti i parametri dell'oggetto fate riferimento alla sezione Capitolo 8 [Oggetti], pagina 13.

Ogni finestra di modifica ha un gadget stringa per il nome dell'oggetto. Questo nome è importante perchè è utilizzato per riferirsi a un determinato oggetto. Notate che attualmente non esiste alcuna possibilità di fare controlli incrociati sui nomi per cui, se cambiate il nome ad un oggetto che è già stato utilizzato da un altro oggetto, il riferimento di quest'ultimo all'oggetto in fase di modifica *non* verrà aggiornato. pertanto, dovrete fare questa modifica a mano.

I gadget a bottone nelle finestre di modifica aprono tipi differenti di requester. Potete scegliere un elemento facendo click e selezionando il gadget **OK**, o potete più semplicemente fare doppio click sull'elemento. Per lasciare un requester senza apportare alcuna modifica utilizzate il gadget **Annulla**. Se volete cancellare un campo che può essere modificato solo dal requester stesso, aprite il requester e selezionate **OK senza selezionare un elemento**.

Le finestre di modifica per i tipi di oggetto Programma e Immagine hanno una caratteristica addizionale: potete infatti lasciare cadere in esse un'icona e impostare i parametri della finestra in accordo con quelli definiti all'interno dell'icona stessa.

## 9.5 Tooltype

Potete impostare diversi tooltype (parametri) nell'icona dell'editor di preferenze in modo da controllarne il comportamento quando lo lanciate dal Workbench.

<b>USE</b>	Se impostate questo tooltype in un'icona per un file di preferenze l'editor considererà il file come quello contenente la configurazione corrente.
<b>SAVE</b>	Se impostate questo tooltype in un'icona per un file di preferenze l'editor considererà il file come quello contenente la configurazione corrente e permanente.
<b>PUBSCREEN</b>	Questo tooltype indica all'editor di aprire le sue finestre su di uno specifico schermo pubblico. Se questo tooltype non viene fornito verrà utilizzato lo schermo pubblico di default.

**CREATEICONS**

Quando questo tootype è impostato a **YES** l'editor creerà un'icona per ogni file di preferenze che viene salvato con il menu **Salva come...**

**DEFAULTFONT**

L'editor normalmente usa il font di default dello schermo pubblico per i suoi gadget. Se impostate a **YES** questo tootype l'editor utilizzerà invece il font di sistema.

**XPOS** Specifica la posizione iniziale X della finestra principale dell'editor.

**YPOS** Specifica la posizione iniziale Y della finestra principale dell'editor.

**MINLISTCOLUMNS**

Questo specifica il numero minimo di colonne nei gadget lista.

**MINLISTROWS**

Questo specifica il numero minimo di righe nei gadget lista.

## 9.6 Argomenti da CLI

Quando l'editor di preferenze è lanciato da una shell (CLI) viene utilizzato lo schema seguente per la riga di comando:

```
FROM,EDIT/S,USE/S,SAVE/S,PUBSCREEN/K,DEFAULTFONT/S
```

**FROM** Questo parametro specifica il nome del file di preferenze che l'editor deve caricare.

**USE** Se usate questo parametro l'editor installerà il file specificato con il parametro **FROM** come il file di configurazione corrente.

**SAVE** Se usate questo parametro l'editor installerà il file specificato con il parametro **FROM** come il file di configurazione corrente e permanente.

**PUBSCREEN**

Questo parametro indica all'editor di aprire le sue finestre su di uno specifico schermo pubblico. Se questo parametro non viene specificato verrà utilizzato lo schermo pubblico di default.

**DEFAULTFONT**

L'editor normalmente usa il font di default dello schermo pubblico per i suoi gadget. Se specificate questo parametro l'editor utilizzerà invece il font di sistema.

## 10 L'interfaccia della libreria di ToolManager

Il cuore di ToolManager è integrato in una libreria shared di Amiga. Questa libreria offre diverse funzioni per la creazione e la manipolazione degli oggetti di ToolManager in modo da poterli utilizzare nei vostri programmi.

Attualmente ci sono solo sei funzioni disponibili:

### `AllocTMHandle()`

Per creare gli oggetti di ToolManager dovete per prima cosa allocare un `TMHandle`. Questo handle memorizza tutte le informazioni relative ai vostri oggetti ed è utilizzato per riferirsi a questi ultimi. Notate che le informazioni conservate in questo handle sono accessibili *solamente* dal programma che lo alloca.

### `FreeTMHandle()`

Questa funzione rilascia un `TMHandle` e tutti gli oggetti di ToolManager ad esso associati. Ogni chiamata alla funzione `AllocTMHandle()` deve essere accoppiata ad una chiamata alla funzione `FreeTMHandle()`!

### `CreateTMOBJECTTags()`

### `CreateTMOBJECTTagList()`

Questa funzione crea un oggetto di ToolManager. Dovete fornire un nome, il tipo di oggetto e vari tag per i parametri dell'oggetto. Il nome dell'oggetto è importante in quanto viene usato per riferirsi all'oggetto stesso.

### `ChangeTMOBJECTTags()`

### `ChangeTMOBJECTTagList()`

Con questa funzione potete modificare i parametri di un oggetto di ToolManager. Lo stato interno dell'oggetto verrà aggiornato con i valori dei nuovi parametri. Nota: al momento non è possibile modificare gli oggetti `Image`.

### `DeleteTMOBJECT()`

Con questa funzione potete cancellare un oggetto di ToolManager. Se l'oggetto è agganciato ad altri oggetti verrà segnalato a questi ultimi di aggiornare il loro stato interno.

### `QuitToolManager()`

Questa funzione indica al gestore di ToolManager di terminare appena possibile.

La descrizione completa dell'interfaccia della libreria è contenuta in un file `AutoDoc` (vedi Paragrafo 7.1 [Docs], pagina 8). (NdT: la documentazione `AutoDoc` non è stata tradotta)

## 11 Come definire un Hot Key

Questo capitolo descrive come definire un Hot Key come una stringa di descrizione dell'input che verrà poi interpretata dalle Commodities Exchange. Ogni volta che viene attivato un Hot Key le Commodities genereranno un evento che è utilizzato da ToolManager per attivare gli oggetti Programma oppure per chiudere o aprire un oggetto Dock. Una stringa di descrizione ha la seguente sintassi:

```
[<classe>] {[<->][<qualificatore>]} [-][upstroke] [<codice tasto>]
```

Tutte le parole chiave possono essere scritte sia in minuscolo che in maiuscolo.

**classe** descrive una classe di evento di input (InputEvent). Questo parametro è opzionale e se non viene specificato verrà utilizzato il valore di default **rawkey**. Vedi Paragrafo 11.1 [Classi di evento], pagina 27.

I qualificatori sono “segnali” che devono essere impostati o meno nel momento in cui il tasto Hot Key è attivato; in caso contrario non verrà generato alcun evento. Per ogni qualificatore che deve essere impostato dovete fornire la sua parola chiave. Se volete ignorare un qualificatore basta far precedere la sua parola chiave dal segno -. Vedi Paragrafo 11.2 [Qualificatori], pagina 27.

Normalmente un evento di Hot Key viene generato alla pressione di un tasto. Se l'evento deve essere generato invece quando il tasto è rilasciato (anziché premuto) specificate la parola chiave **upstroke**. Quando sia la pressione che il rilascio di un tasto devono generare un evento specificate **-upstroke**.

Il codice tasto è dipendente dalla classe di evento. Vedi Paragrafo 11.3 [Codici tasto], pagina 28.

Nota: scegliete *con cura* i vostri Hot Key poichè le Commodities hanno una priorità alta nella catena dei gestori di input (in altre parole, gli Hot Key da voi definiti saranno sovrapposti ad altri precedentemente definiti).

### 11.1 Classi di evento

Le Commodities supportano molte delle classi di evento (InputEvent) generate dall'input.device. Questa sezione descrive quelle classi che sono più utili per gli Hot Key di ToolManager.

- rawkey** Questa è la classe di default e ricopre tutti gli eventi di tastiera. Per esempio **rawkey** a o a genera un evento ogni volta che viene premuto il tasto “a”. Dovete specificare un codice tasto per questa classe. Vedi Paragrafo 11.3.1 [Codici tasto per rawkey], pagina 28.
- rawmouse** Questa classe descrive tutti gli eventi relativi ai bottoni del mouse. Dovete specificare un codice tasto per questa classe. Vedi Paragrafo 11.3.2 [Codici tasto per rawmouse], pagina 29.
- diskinserted**
- Gli eventi di questa classe sono generati quando un disco viene inserito in un drive. Questa classe non ha alcun codice tasto.
- diskremoved**
- Gli eventi di questa classe sono generati quando un disco viene estratto da un drive. Questa classe non ha alcun codice tasto.

## 11.2 Qualificatori

Alcuni sinonimi delle parole chiave sono stati aggiunti nelle Commodities V38. Questi sono evidenziati da \*.

**lshift, left\_shift \***

Tasto Shift sinistro.

**rshift, right\_shift \***

Tasto Shift destro.

**shift** Uno dei due tasti Shift.

**capslock, caps\_lock \***

Tasto Caps Lock.

**caps** Uno dei due tasti Shift o il tasto Caps Lock.

**control, ctrl \***

Tasto Control.

**lalt, left\_alt \***

Tasto Alt sinistro.

**ralt, right\_alt \***

Tasto Alt destro.

**alt** Uno dei due tasti Alt.

`lcommand`, `lamiga *`, `left_amiga *`, `left_command *`

Tasto Amiga/Command sinistro.

`rcommand`, `ramiga *`, `right_amiga *`, `right_command *`

Tasto Amiga/Command destro.

`numericpad`, `numpad *`, `num_pad *`, `numeric_pad *`

Questa parola chiave *deve* essere usata per ogni tasto presente sul tastierino numerico.

`leftbutton`, `lbutton *`, `left_button *`

Tasto mouse sinistro. Vedi note più sotto.

`midbutton`, `mbutton *`, `middlebutton *`, `middle_button *`

Tasto mouse centrale. Vedi note più sotto.

`rbutton`, `rightbutton *`, `right_button *`

Tasto mouse destro. Vedi note più sotto.

**repeat** Questo qualificatore è impostato quando è attiva la ripetizione dei tasti. Utile solo nei casi di classe di evento **rawkey**.

Nota: le Commodities V37 hanno un bug che inibisce l'uso dei qualificatori **leftbutton**, **midbutton** e **rbutton**. Questo bug è stato corretto nella V38.

## 11.3 Codici tasto

Ogni classe di evento (`InputEvent`) ha i suoi codici tasto:

### 11.3.1 Codici tasto per la classe di evento **rawkey**

Alcuni sinonimi delle parole chiave sono stati aggiunti nelle Commodities V38. Questi sono evidenziati da **\***.

**a-z**, **0-9**, ...

Caratteri ASCII.

**f1**, **f2**, ..., **f10**, **f11 \***, **f12 \***

Tasti funzione.

```
up, cursor_up *, down, cursor_down *
left, cursor_left *, right, cursor_right *
```

Tasti cursore.

```
esc, escape *, backspace, del, help
tab, comma, return, space, spacebar *
```

Tasti speciali.

```
enter, insert *, delete *
page_up *, page_down *, home *, end *
```

Tasti del tastierino numerico. Ognuno di questi codici tasto *deve* essere usato con il qualificatore `numericpad`!

### 11.3.2 Codici tasto per la classe di evento `rawmouse`

Queste parole chiave sono state aggiunte nelle Commodities V38 e non sono pertanto disponibili nella V37.

```
mouse_leftpress
```

Pressione del tasto sinistro del mouse.

```
mouse_middlepress
```

Pressione del tasto centrale del mouse.

```
mouse_rightpress
```

Pressione del tasto destro del mouse.

Nota: per usare uno di questi codici tasto dovete anche specificare il corrispondente qualificatore, per esempio:

```
rawmouse leftbutton mouse_leftpress
```

## 11.4 Esempi di Hot Key

```
ralt t    Premere contemporaneamente il tasto Alt destro e il tasto "t"
```

```
ralt lalt t
```

Premere contemporaneamente i tasti Alt sinistro e destro e il tasto "t"

`alt t` Premere contemporaneamente uno dei tasti Alt e il tasto “t”

`rcommand f2`

Premere contemporaneamente il tasto Amiga destro e il secondo tasto funzione

`numericpad enter`

Premere il tasto Enter sul tastierino numerico

`rawmouse midbutton leftbutton mouse_leftpress`

Premere contemporaneamente i tasti sinistro e centrale del mouse

`diskinserted`

Inserire un disco in un drive



## Appendice A Le domande più frequenti su ToolManager

Ecco le risposte alle domande poste più frequentemente su ToolManager:

- Perchè ToolManager non può creare menu “Strumenti” multipli oppure sotto-menu?

Menu multipli o sotto-menu non sono attualmente supportati dal software di sistema. Per crearli dovrete *forzarli* all’interno del software di sistema, il che può portare ad un sistema instabile. Dal momento che non voglio rendere instabile il vostro sistema, non implementerò una cosa del genere in ToolManager.

- I programmi WB non partono, ma tutti gli altri tipi di programmi sì.

ToolManager si basa sul programma `L:WBStart-Handler` per lanciare i programmi WB. Ci possono essere due ragioni per le quali ToolManager non è in grado di eseguire questo programma:

Il file ‘`L:WBStart-Handler`’ non esiste. Copiatelo dall’apposito archivio di distribuzione.

Il flag di esecuzione (e) non è stato impostato per questo file. Usate il seguente comando:  
`protect L:WBStart-Handler +e`

- Come posso creare una finestra dock orientata orizzontalmente?

Impostate il numero di colonne uguale al numero di tool dell’oggetto Dock.

- Come posso creare una finestra di output per i programmi CLI?

Le finestre di output possono essere create utilizzando il device `CON:`. Usate il seguente nome di file per creare una finestra ad apertura automatica con un gadget di chiusura (e che non si chiuderà dopo che il programma è terminato):

`CON:10/10/640/100/Output-Window/AUTO/CLOSE/WAIT`

Il device `CON:` ha diverse opzioni, consultate il vostro manuale AmigaDos per maggiori informazioni.

- Come posso inserire gli argomenti nel mezzo di una linea di comando CLI/ARexx ?

Normalmente tutti gli argomenti vengono accodati alla linea di comando. Per inserire gli argomenti in un qualsiasi punto della linea di comando ToolManager usa la stessa sintassi utilizzata dal comando `alias` della AmigaShell. Così, ad esempio,

`Dir [] all`

inserirà tutti gli argomenti prima della parola chiave `all`.

- Come posso cancellare un riferimento ad un oggetto semplice in un oggetto complesso?

Dopo aver premuto il bottone di selezione dell’oggetto premete il bottone “OK” *senza* aver selezionato un oggetto. Questo indicherà che non avete scelto un oggetto e quindi il riferimento verrà cancellato.

- Come posso creare dei sotto-dock?

Dovete utilizzare oggetti Programma del tipo Dock. Mettete questi oggetti negli elementi del vostro dock principale e questi apriranno o chiuderanno gli altri dock.

- ToolManager non risponde più dopo aver lanciato un comando di rete.

Esiste attualmente un problema con il software di rete a causa del quale le richieste locali non hanno un tempo limite massimo (timeout). Se per esempio la vostra macchina si chiama **Macchina1** e avete un oggetto Programma del tipo Network che attiva il comando **Oggetto@Macchina1**, ToolManager entrerà in un loop senza fine quando lo attivate. Usate solo i nomi di macchine remote!

## Appendice B La storia di ToolManager

NdT: questo capitolo non è stato tradotto su richiesta dell'autore.

### 2.1, Release date 16.05.1993

- New Exec object types: Dock, Hot Key, Network
- New Dock object flags: Backdrop, Sticky
- New object type: Access
- Network support
- Editor main window is now an AppWindow
- Gadget keyboard shortcuts in the preferences editor
- New tooltypes for the preferences editor
- Several bug fixes
- Enhanced documentation

### 2.0, Release date 26.09.1992, Fish Disk #752

- Complete new concept (object oriented)
- (Almost) Complete rewrite
- ToolManager is now split up into two parts
- Main handler is now embedded into a shared library
- Configuration is now handled by a Preferences program
- Configuration file format has changed again :-) It is an IFF File now and resides in ENV:
- Multiple Docks and multi-column Docks
- Docks with new window design
- Dock automatically detects largest image size
- Sound support
- Direct ARexx support for Exec objects
- ToolManager can be used without the Workbench. If the Workbench isn't running, it won't use any App\* features.
- Locale support
- Path from Workbench will be used for CLI tools
- Seperate Handler Task for starting WB processes

### 1.0 upto 1.5

Please look into the english documentation for further information.

## Appendice C L'autore ringrazia. . .

ToolManager è passato attraverso molte fasi di evoluzione sin dalla sua prima implementazione uscita a metà del 1990. Questo sviluppo non sarebbe stato possibile se non avessi ricevuto molti suggerimenti da vari utenti di ToolManager. Molte idee e caratteristiche provengono da questi contributi. . .

Di conseguenza vorrei ringraziare:

Per il test Alpha/Beta, idee e segnalazione di bug:

La sezione Amiga del nostro computer club (Computerclub an der RWTH Aachen), Olaf 'Olsen' Barthel, Georg Hessmann (Gucky), Markus Illenseer (ill), Klaus Melchior, Rickard Olsson (Richie), Matthias Scheler (Tron), Ralph Schmidt (laire), Roger Westerlund (Budda), Juergen Weinelt, Brian Wright (SteveVai), Petra Zeidler (stargazer) e molti altri. . .

Matthew Dillon

Senza il suo **eccellente** sistema di sviluppo DICE C e altri suoi tool ToolManager non esisterebbe!

Per l'eccellente lavoro grafico:

Andreas Harrenberg, Georg Hessmann, Michael "Mick" Hohmann, Markus Illenseer, Oliver Koenen, Klaus Melchior, Rickard Olsson, Jan Peter, Matthias Scheler, Brian Wright

Per le traduzioni:

Tomi Blinnikka (suomi), Jorn Halonen (norsk), Dr. Peter Kittel (deutsch), Jasper Kehlet (dansk), Klaus Melchior (eifel), Rickard Olsson (svenska), Rullier Pascal (français), Marc Schaefer (français), Tor Rune Skoglund (norsk), Reinhard Spisser (italiano), Andrea Suatoni (italiano)

Tutti il personale a West Chester:

Per lo sviluppo di Amiga e il suo superbo sistema operativo.

Tutti gli utenti che mi hanno spedito del denaro:

Io non lo chiesi per le release 1.X ma è bello vedere che qualcuno apprezza il tuo lavoro.

Tutti gli utenti che mi hanno mandato un messaggio:

Ho realmente apprezzato la lettura delle vostre lettere!

e tutti quelli che mi sono dimenticato di nominare. . .

# Indice

## A

AmigaGuide .....	8
Argomenti da CLI .....	24

## B

Bug report .....	2
------------------	---

## C

Caratteristiche V38 (e successive) .....	1
Classi InputEvent .....	27
Codice sorgente .....	11
Codici tasto per <b>rawkey</b> .....	28
Codici tasto per <b>rawmouse</b> .....	29
Commenti .....	2
Concetti .....	5
Concetti del programma .....	5
Configurazione .....	21

## D

DeleteTool .....	9
Directory Docs .....	8
Directory Goodies .....	9
Directory Graphics .....	9
Directory L .....	10
Directory Libs .....	10
Directory Locale .....	10
Directory Prefs .....	11
Directory Programmers .....	11
Directory Scripts .....	11
Directory Source .....	11
Directory WBStartup .....	12
Diskinserted .....	27
Diskremoved .....	27
Documentazione .....	8
Documentazione ASCII .....	8
Documentazione della libreria .....	8
Documentazione stampata .....	8
Domande .....	31
Donazioni .....	2

## E

E-Mail .....	2
Editor di preferenze .....	21
Esempi di Hot Key .....	29
Esempio .....	6

## F

File dei catalog .....	10
File delle lingue .....	10
File di distribuzione .....	8

## G

GetPubName .....	9
GiftWare .....	1

## I

Immagini aggiunte .....	9
Immagini di esempio .....	9
Indirizzi .....	2
Indirizzo InterNet .....	2
Indirizzo postale .....	2
Installazione (veloce) .....	3
Installazione rapida .....	3
Interfaccia della libreria .....	25
Interfaccia della libreria shared .....	25
Introduzione a ToolManager .....	4
Introduzione agli Hot Key .....	26
Introduzione agli oggetti .....	5

## L

Lingue .....	10
Lista: codici tasto per <b>rawkey</b> .....	28
Lista: codici tasto per <b>rawmouse</b> .....	29
Lista: Qualificatori .....	27
Localizzazione .....	10

## N

Note importanti .....	1
-----------------------	---

**O**

Oggetti .....	13
Oggetti accesso .....	19
Oggetti di ToolManager .....	13
Oggetti Dock .....	17
Oggetti Icona .....	16
Oggetti immagine .....	14
Oggetti menu .....	15
Oggetti programma .....	13
Oggetti suono .....	15

**P**

Player di suoni .....	9
-----------------------	---

**Q**

Qualificatori .....	27
---------------------	----

**R**

Rawkey .....	27
Rawmouse .....	27
Riferimento: Editor di preferenze .....	21
Riferimento: File di distribuzione .....	8
Riferimento: Hot Key .....	26
Riferimento: Interfaccia libreria .....	25
Riferimento: Oggetti di ToolManager .....	13
Ringraziamenti .....	34

Risposte .....	31
----------------	----

**S**

Script ARexx .....	11
Script Shell .....	11
Storia .....	33
Supporto per i compilatori .....	11

**T**

TeX .....	8
Texinfo .....	8
Tooltype .....	23
Tour guidato .....	6
Traduzioni .....	10, 11
Tutorial .....	6

**U**

UPD .....	9
-----------	---

**V**

Versioni .....	33
Versioni del programma .....	33

**W**

WBStart 1.2 .....	10
WBStart-Handler .....	10

# Indice

<b>1</b>	<b>Note importanti .....</b>	<b>1</b>
<b>2</b>	<b>Dove mandare bug reports, commenti e donazioni .....</b>	<b>2</b>
<b>3</b>	<b>Come installare rapidamente ToolManager 2.1 ....</b>	<b>3</b>
<b>4</b>	<b>Che cosa è ToolManager?.....</b>	<b>4</b>
<b>5</b>	<b>I concetti alla base di ToolManager.....</b>	<b>5</b>
<b>6</b>	<b>Un tour guidato attraverso ToolManager .....</b>	<b>6</b>
<b>7</b>	<b>Descrizione di tutti i file nella distribuzione .....</b>	<b>8</b>
7.1	La directory Docs .....	8
7.2	La directory Goodies .....	9
7.3	La directory Graphics .....	9
7.4	La directory L .....	10
7.5	La directory Libs.....	10
7.6	La directory Locale .....	10
7.7	La directory Prefs.....	11
7.8	La directory Programmers.....	11
7.9	La directory Scripts .....	11
7.10	La directory Source .....	11
7.11	La directory WStartup .....	12
<b>8</b>	<b>Manuale di riferimento degli oggetti di ToolManager .....</b>	<b>13</b>
8.1	Oggetti programma .....	13
8.2	Oggetti immagine .....	14
8.3	Oggetti suono .....	15
8.4	Oggetti menu .....	15
8.5	Oggetti Icona .....	16
8.6	Oggetti Dock .....	17
8.7	Oggetti accesso.....	19

<b>9</b>	<b>L'editor di preferenze di ToolManager .....</b>	<b>21</b>
9.1	Gadget della finestra principale .....	21
9.2	Menu finestra principale .....	22
9.3	Finestra di creazione oggetti .....	22
9.4	Finestre di modifica .....	23
9.5	Tooltype .....	23
9.6	Argomenti da CLI .....	24
<b>10</b>	<b>L'interfaccia della libreria di ToolManager .....</b>	<b>25</b>
<b>11</b>	<b>Come definire un Hot Key .....</b>	<b>26</b>
11.1	Classi di evento .....	26
11.2	Qualificatori .....	27
11.3	Codici tasto .....	28
11.3.1	Codici tasto per la classe di evento <code>rawkey</code> .....	28
11.3.2	Codici tasto per la classe di evento <code>rawmouse</code> .....	29
11.4	Esempi di Hot Key .....	29
<b>Appendice A</b>	<b>Le domande più frequenti su ToolManager .....</b>	<b>31</b>
<b>Appendice B</b>	<b>La storia di ToolManager .....</b>	<b>33</b>
<b>Appendice C</b>	<b>L'autore ringrazia .....</b>	<b>34</b>
<b>Indice .....</b>		<b>35</b>