

IDerPrefs

Ian Campbell

Copyright © CopyrightÂ©1998 Ian Campbell

COLLABORATORS

	<i>TITLE :</i> IDerPrefs		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Ian Campbell	April 18, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	IDerPrefs	1
1.1	IDerPrefs v1.0	1
1.2	Introduzione	1
1.3	Distribuzione	2
1.4	Requisiti di sistema	2
1.5	Caratteristiche	3
1.6	Installazione	3
1.7	Configurazione	3
1.8	Modo facile	4
1.9	tecnico	5
1.10	Modo esperto	5
1.11	Modo esperto	7
1.12	Informazioni sull'autore	7
1.13	Grazie a tutti	8
1.14	Storia Cronologica del Programma	8

Chapter 1

IDerPrefs

1.1 IDerPrefs v1.0

IDerPrefs 1.0 Editor File di Configurazione per IDer ←
1.06

Copyright © 1998 Ian Campbell

~Introduzione~~~~~

~Distribuzione~~~~~

Sistema richiesto

Caratteristiche

Installazione

Configurazione

Modo facile

Modo esperto

Informazioni autore

Ringraziamenti

Cronologia

1.2 Introduzione

Introduzione

Gli utenti che usano IDer, sanno bene come questo sia un modo molto più facile per visualizzare dati. Basta impostare IDer come programma associato

in tutte le proprie icone progetto e non è più necessario sapere il tipo di file per il quale viene creata l'icona: un doppio click su di essa richiamerà il programma adatto per visualizzare quel file. Non si verificheranno così più casi di programmi diversi per icone diverse, percorsi sbagliati ecc. La cosa più importante è che non si tratti di un hack. E' solo un modo alternativo di pensare ed utilizzare il sistema operativo.

Tutte queste funzioni si basano sul file "IDer.config" che quantunque non sia particolarmente complesso, siamo ormai nel 1998 e non dovresti ridurci ancora a modificare a mano un file di testo per la configurazione di un programma.

IDerPrefs

Ecco quindi lo scopo di IDerPrefs. E' semplicemente la "faccia graziosa" di IDer con in più alcune funzioni specifiche che non si possono trovare in un text editor. (Sì, forse su Emacs, ma quello non è un text editor, è una nuova forma di vita).

Ora ogni volta che scarichi un nuovo ultimo-velocissimo-visualizzatore-jpeg o un riproduttore-sonoro-32bit-super-duper, ti basterà estrarre un'icona per visualizzare tutti i dati relativi presenti sul tuo sistema con il nuovo programma.

Questo è particolarmente utile mentre stai navigando in qualche CD, e infatti inizialmente IDerPrefs era una specie di un CUCDPrefs ma per un uso più generico su HD ecc. (CUCD prefs funziona solo se hai un CUCD dal momento che devi avviare InitCD) Brilliant CD a proposito di CU-Amiga.

1.3 Distribuzione

Disclamazione

Utilizzando questo prodotto, ti prendi la piena responsabilità per qualsiasi danno o perdita conseguente al suo uso. Io (Ian Campbell) non ne sono responsabile.

Distribuzione

IDerPrefs è freeware. Non ti costa assolutamente nulla.

Questo pacchetto può essere distribuito liberamente tramite BBS; Internet (urrà!); Collezioni di Software come Aminet (doppio urrà!); e CD-ROM purché l'archivio rimanga intatto e non venga modificato nessuno dei file.

1.4 Requisiti di sistema

Sistema Richiesto

- Kickstart 3.0 o superiore
- Processore 68020 o superiore
- MUI 3.8 o superiore (util/libs/mui38usr.lha)
(Può funzionare anche su una versione precedente, ma deve essere

- almeno la 2.0 perché usa le classi custom)
- IDer 1.06 (util/wb/ider.lha)

E' stato programmato su un A1200 3.1 e MUI Demo 3.8 perciò dovrebbe funzionare perlomeno su questa configurazione.

1.5 Caratteristiche

Caratteristiche

- Interfaccia MUI, che ti permette la personalizzazione della GUI secondo i tuoi gusti e verso la tua "unica" idea di stile. (qualunque essa sia) <gulp!>
- Uso della tecnica di drag & drop per la creazione automatica di nuove classi tramite trascinamento di un datatype.
- 'Modo esperto' che ti permette di editare ogni più piccolo dettaglio del file di configurazione.
- 'Modo facile' per l'uso quotidiano.
- Tutti i pulsanti/menu ecc. funzionano come tutti gli altri editor di preferenze, perciò ora potresti anche uscire da questa guida.
- Aiuto in linea sensibile al contesto e fumetti di aiuto.
- Caricamento parziale di file di configurazione corrotti.
- Permette di "Resettare a predefinito" i file di configurazione.
- Completamente localizzato.

1.6 Installazione

Installazione

Usa lo script di installazione oppure trascina semplicemente l'icona ovunque vuoi e imposta (opzionalmente) il tootype

HELPPFILE

per avere

l'aiuto in linea e il tootype

ORIGFILE

per avere l'opzione

Resetta a predefinito.

1.7 Configurazione

Attualmente IDerPrefs può essere eseguito solo da Workbench e può essere configurato attraverso i seguenti tootype (tutti opzionali).

SAVENAME - Questo è il nome/percorso dove bisogna salvare il file di configurazione alla pressione del pulsante 'Salva'.
Predefinito: "envarc:IDer.prefs"

USENAME - Questo è il nome/percorso dove bisogna salvare il file di configurazione alla pressione del pulsante 'Usa'.
Predefinito: "env:IDer.prefs"

- ORIGNAME - Questo è il nome/percorso da dove caricare il file di configurazione alla selezione dell'opzione "Resetta a predefinito" dalla voce di menu. Come predefinito è una copia del file di ←
configura-
zione che viene distribuito con IDer ma può essere qualunque altro file. Per esempio, se stessi usando IDer + IDerPrefs per i dati contenuti su un CD-ROM e volessi modificare i programmi di visualizzazione per la tua configurazione, potresti usare questa opzione per utilizzare un file di configurazione ←
esterno,
contenuto sul CD, di tuo gradimento nel caso che questi lo chiudano ←
.
- HELPNAME - Questo è il nome/percorso del file di aiuto. Se specificato, verrà caricato alla pressione del tasto Help.
Predefinito: "progdir:IDerPrefs.guide"
- EXPERTMODE - (=YES|NO) Avvia il programma nel 'Modo esperto'.
Il modo 'La terribile verità'. Per default è impostato come NO. (il modo 'Beata Ignoranza-Non so niente-Spegni gli X-Files-LaLaLaLa non posso ascoltarti).
- CREATEICONS - (=YES|NO) Se per qualche strana ragione ti piace salvare tanti piccoli file di configurazione un po' dappertutto con la funzione 'Salva come', questo tooltype ti aiuterà a tenerne traccia.
Con i pulsanti 'Salva' e 'Usa' non avrai mai delle icone.
Predefinito: YES (per coerenza con gli editor di sistema).

1.8 Modo facile

Modo facile

E' veramente molto facile.

Il nome sopra lo slider - mostra il nome del file di configurazione che è stato caricato all'avvio del programma, e probabilmente è quello originale. Non è il nome del file che viene salvato con il pulsante 'Salva', che viene definito dal tooltype

SAVENAME

ed è quello predefinito.

Se vuoi sapere il nome del file che verrà salvato, lascia fermo per qualche ←
secondo

il puntatore del mouse sopra il pulsante 'Salva'.

Il pulsante elimina - elimina la voce mostrata correntemente. Se provi ad eliminare l'ultima voce, otterrai un file di configurazione che avrà come opzione di default solo Multiview. Molto eccitante.

Trascinare un programma - in qualsiasi punto dell'interfaccia tranne che sullo slider, imposta l'Azione per la classe corrente con il nome del file trascinato.

Trascinare un descrittore dataType sullo slider creerà automaticamente

una nuova voce, basata sulla descrizione file contenuta nel datatype.
Vedi: tecnico

Salva, usa, annulla - Fanno quello che fanno sempre. In questo caso puoi specificare i nomi dei file da salvare che verranno uniti a questi pulsanti ↔

Vedi:

```

    configurazione
    . Ricorda che il pulsante 'Salva'
salverà sempre il SAVENAME e il USENAME, (p.e. sia in env: che in envarc:).
```

1.9 tecnico

Quando trascini l'icona di un Datatype sullo slider, IDerPrefs esamina il datatype stesso e crea una nuova voce per te. Questo è buono perché oramai vi sono decine e decine di ottimi datatype per tutti i gusti.

Cosa Legge il Parser:

Il campo CLASSE: viene costruito concatenando dth_Name con dth_GroupID
p.e. GIF + Immagine = GIF_Immagine; 8SVX + Suono = SVX_Suono.

Il campo NOME: Se il DTD contiene un nome file reg expn (??) p.e. #?.(jpg|jpeg ↔
)
verrà posizionato nel campo Nome.

Il campo STRINGA: Se il DTD contiene una maschera del file, questa verrà inserita nel campo Offset con in testa uno "0,". Verrà inserita anche come maschera esadecimale se il DTD ha un bit set sensibile alle maiuscole o come maschera ASCII (tra virgolette) se è unset. La sola eccezione che ho trovato a questa assunzione è il datatype CDXL che contiene una comparazione non sensibile alle maiuscole?! (Penso sia un pessimo formato).

Il sottocampo FLAGS=I: Il valore di questi bit sensibili alle maiuscole contenuti nei datatype saranno usati anche per impostare il bit di IDer FLAGS=I. (Al valore opposto).

Il campo AZIONE: (futuristico :) Se il datatype contiene uno o più chunk DTTL (Datatype Tool) e uno di questi è di tipo '3' (=editor tool), verrà messo direttamente nel campo Azione. Se non riesce proverà a inserirvi il tool tipo '2' (=browser). Non sono a conoscenza di nessun datatype che attualmente sfrutti questa caratteristica, nemmeno l'ottimo akdatatype, che avrei pensato sarebbe un'ottima opportunità per pubblicizzare SuperView!

Datatypes con codice hook di identificazione proprietari:

Se il datatype usa un hook di comparazione proprietario per identificare i file (un esempio di questo è il datatype MacSnd), verrà respinto da IDerPrefs (per ovvie ragioni).

1.10 Modo esperto

Modo esperto

Anche questo è veramente molto facile. (Infatti se hai visto il sorgente avrai notato che entrambi i modi sono una sottoclasse di qualcosa chiamato "EZDisplay").

Se hai già modificato un file di configurazione di IDer a mano, puoi leggere più velocemente

questo

.

Con il modo esperto hai a disposizione una nuova sezione, situata nella parte più bassa dell'interfaccia, che ti permette di modificare le varie classi.

Anche la sezione superiore ha delle nuove opzioni: ora c'è sia un pulsante di inserimento che di eliminazione, e il nome della classe può essere modificato tramite un gadget stringa.

Inserisci - Crea una nuova voce, vuota, per la modifica manuale.

Stack: - La dimensione dello stack per il programma contenuto nel campo Azione. Come predefinito è di 4096 bytes.

Pri: - Come sai Amiga è multitasking :) Questa è la priorità del programma, da -127 a 128. Come predefinito è 0.

No virgolette nome file - Attiva questa voce se il tuo (vecchio) programma non riesce ad esaminare i nomi dei file tra virgolette, altrimenti IDer racchiuderà il nome del file tra virgolette per poter inserire degli spazi tra le parole.

Modo Esamin.: Ciclico - IDer utilizza 3 modi per identificare un file (suffisso, nome file e stringhe di caratteri contenuti in qualche punto del file). Quanti di questi modi devono essere utilizzati prima di eseguire il programma di visualizzazione?

Nome: - Questa è una normale espressione AmigaDOS, come quelle che si vedono sempre nelle shell. p.e. (mod.#?|#?.mod)

Suffisso: - Se non hai bisogno della potenza offerta dalle opzioni Nome, puoi usare anche questa funzione, per esempio .GIF

Stringa: - Questa è l'opzione più potente. Sapevi che la maggioranza dei file attuali ha una specie di identificatore interno al file che dicono di che tipo sono? Se conosci questo identificatore puoi inserirlo qui.
Usa: <offset>, <valore> per un numero magico (ed esprimilo con la notazione esadecimale per esempio 16,DE..BE.. (dove <..> è una wildcard).
Usa: <offset>, "stringa" se c'è una stringa stampabile che sai essere sempre presente in ogni file di quel tipo.
Per esempio, i documenti AmigaGuide iniziano sempre con la stringa "@database", quindi potresti inserire 0,"@database". (in questo caso la wildcard è <?>).

Questo è il modo di esame più efficace, perché funziona anche se, per esempio, un file jpeg viene nominato con suffisso .gif! Questo campo viene quasi sempre riempito se viene trascinato l'icona di un datatype.

Insensibile alle maiuscole - Se stai usando il modo di esame Stringa, dovresti attivare questo parametro, senno puoi anche farne a meno. Con questa impostazione, digitare @Database, @dataBase o ancora, @DaTaBaSE (per esempio in alcuni file .guide di pirati software :) non farà alcuna differenza.

Non ti preoccupare, non devi imparare a memoria tutte queste cose, perché ci sono i fumetti di aiuto.

1.11 Modo esperto

Ero un editor di testo...

Tutti i nomi dei gadget stringa, corrispondono esattamente ai loro parametri equivalenti nel file di configurazione tranne:

Il campo "Stringa" che contiene ciò che è inserito come "OFFSET=".

L'opzione "Insensibile al contesto" che attiva il flag "I".

L'opzione "No Virgolette Nome File" che attiva il flag "Q".

Il gadget ciclico "Modo Esamin." che attiva i flag "A" e "T".

1.12 Informazioni sull'autore

Informazioni sull'Autore

Sono un ragazzo di 23 anni e vivo a Trentham, Upper Hutt nell'Isola del Nord della Nuova Zelanda, dove i bellissimi animali nativi vivono indisturbati sulle montagne innevate e l'aria è pulita, ma, ancora più importante, un accesso ad Internet completo di tutto costa solo 40\$ al mese (=13 GBP), le telefonate urbane sono libere e MTV Europe è una stazione TV libera-nell'aria! Ho anche una laurea in computer science alla Victoria University di Wellington.

Mi sono dedicato ad Amiga da quando ho venduto il mio C64 nel 1989 ed ho comprato un'Amiga 500. Il mio sistema attuale è un A1200 con Blizzard '030/50, 16+2 RAM, CDROM6x, Monitor C=1942 e due drive da 2.5" stipati all'interno - uno da 500MB e uno da 60MB. (Forse l'immagine del sistema Amiga "medio" in questi giorni o è lo '040 lo standard ora?)

IDerPrefs è il mio primo programma AmigaOS, ed è stato scritto semplicemente per distrarmi dall'incombente domanda attuale "Cosa dovrei fare della mia vita?", siccome sono appena tornato da un anno di vacanza di lavoro in Inghilterra e sto già programmando il mio prossimo trasloco. Perciò ti prego di spedirmi molti bug reports/suggerimenti così da avere una scusa per
continuare
a rimandare!

EMail

icampbell@extra.co.nz

Per tutti quelli a cui interessa, IDerPrefs è stato scritto in stile semi-O/O in C++ (tutti gli accessi ai file vengono effettuati attraverso delle classi) e compilato grazie al SAS/C 6.58.

1.13 Grazie a tutti

Voglio ringraziare gli autori dei seguenti programmi:

- "IDer 1.06" di Dave Dustin (ovviamente). (util/wb/IDer.lha)
- "SAS/C 6.58" del SAS Institute ma specialmente a Steve Krueger.
- "MUI 3.8" di Stefan Stuntz. (util/libs/mui38demo.lha)
- "Localizer 1.37" di Massimiliano Origgi che effettuato la localizzazione del programma in un battibaleno. (dev/misc/Localizer1.37)
- "MakeDT 1.3" di Michal Letowski, interi codici sorgente AREXX mi hanno ← permesso di scrivere una classe C++ per quei piccoli file datatype in devs:datatypes (dev/misc/MakeDT-1.3.lha)
- I curatori di Aminet
- Davide Romanini e Francesco Celli che hanno tradotto il programma in ← italiano

1.14 Storia Cronologica del Programma

Versione 1.0 (28.1.1998)

- Prima versione pubblica.