

**scout-39**

**COLLABORATORS**

	<i>TITLE :</i> scout-39		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 27, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>scout-39</b>	<b>1</b>
1.1	scout-39.guide	1
1.2	scout-39.guide/Introduction	2
1.3	scout-39.guide/Copyright	2
1.4	scout-39.guide/System Requirements	4
1.5	scout-39.guide/MUI	5
1.6	scout-39.guide/Identify	5
1.7	scout-39.guide/AmiTCP	6
1.8	scout-39.guide/Installation	6
1.9	scout-39.guide/Using Scout	6
1.10	scout-39.guide/Allocations	8
1.11	scout-39.guide/Assigns	9
1.12	scout-39.guide/BoopsiClasses	10
1.13	scout-39.guide/Commodities	11
1.14	scout-39.guide/Devices	13
1.15	scout-39.guide/Expansions	14
1.16	scout-39.guide/Fonts	16
1.17	scout-39.guide/InputHandlers	17
1.18	scout-39.guide/Interrupts	19
1.19	scout-39.guide/Libraries	20
1.20	scout-39.guide/Locks	22
1.21	scout-39.guide/LowMemory	23
1.22	scout-39.guide/Memory	24
1.23	scout-39.guide/Mounted Devs	25
1.24	scout-39.guide/Ports	26
1.25	scout-39.guide/Resident Cmds	27
1.26	scout-39.guide/Residents	28
1.27	scout-39.guide/Resources	29
1.28	scout-39.guide/ScreenMode	30
1.29	scout-39.guide/Semaphores	32

---

---

1.30	scout-39.guide/System	33
1.31	scout-39.guide/Tasks	33
1.32	scout-39.guide/Timer	36
1.33	scout-39.guide/Vectors	37
1.34	scout-39.guide/Windows	38
1.35	scout-39.guide/Scout and AmiTCP	39
1.36	scout-39.guide/Scout without MUI	40
1.37	scout-39.guide/Options	40
1.38	scout-39.guide/Commands	43
1.39	scout-39.guide/Updates	51
1.40	scout-39.guide/Credits	51
1.41	scout-39.guide/Author Info	52
1.42	scout-39.guide/Index	52

---

# Chapter 1

## scout-39

### 1.1 scout-39.guide

Scout 37.204

Release 2.10

Benutzer Handbuch

Copyright (C) 1994-97 Andreas Gelhausen, Richard Körber

Introduction

Was ist Scout?

Copyright

Bitte lesen Sie dies zuerst!

System Requirements

Was zum Betrieb nötig ist

Installation

Wie installiert man Scout?

Using Scout

Wie benutzt man Scout?

Scout and AmiTCP

Was Scout für AmiTCP bietet

Scout without MUI

MUI ist nicht unbedingt erforderlich!

Options

Liste von benutzbaren Shell-Optionen

Commands

Befehle via ARexx und Shell

---

Updates	Wie und wo bekommt man Updates?
Credits	Wem ich zu danken habe
Author Info	Wie erreicht man den Autor?
Index	Stichwortverzeichnis

## 1.2 scout-39.guide/Introduction

Was ist Scout?

=====

Scout ist ein Systemmonitor, d.h. viele für den reibungslosen Betrieb des Rechners notwendige Strukturen -- wie z.B. Tasks, Ports, Assigns, System-Erweiterungen, residente Befehle, Interrupts, usw. -- können angeschaut und auf viele dieser Strukturen können auch bestimmte Aktionen ausgeführt werden.

Es können zum Beispiel Tasks und Prozesse eingefroren, Windows und Screens geschlossen, Semaphore freigegeben und Interrupts aus dem System entfernt werden.

Scout bietet zusätzlich die Möglichkeit, via AmiTCP auch andere Rechner beobachten und gegebenenfalls auch dort auf viele Strukturen zugreifen zu können.

Fast alle der implementierten Funktionen stehen auch als Shell-Parameter zur Verfügung. Das Magic User Interface ist nur für die grafische Benutzungsoberfläche notwendig und demnach nicht unbedingt erforderlich.

## 1.3 scout-39.guide/Copyright

Copyright

=====

Bitte lesen Sie sich diesen Teil aufmerksam durch! Sie akzeptieren diese Bedingungen durch den Start der Software, selbst für einen Probelauf.

COPYRIGHT

Scout 37.204 (Release 2.10) - Copyright (C) 1994-97 by Andreas

---

Gelhausen und Richard Körber, alle Rechte vorbehalten.

Diese Software ist urheberrechtlich geschützt. Die Urheber sind Andreas Gelhausen und Richard Körber.

Sie haben nur das Recht, diese Software zu benutzen, aber keine Rechte an der Software an sich. Disassemblieren, Ressourcen und alle anderen Arten des Reverse Engineering sind verboten.

#### FREEWARE

Scout ist FreeWare. Sie dürfen dieses Paket verwenden, ohne eine Gebühr oder ähnliches an die Autoren zahlen zu müssen. Natürlich freuen wir uns über eine kleine Spende! ;-)

#### KOPIEREN

Sie dürfen dieses Paket so lange kopieren, wie es vollständig und unverändert bleibt.

Sie dürfen das Paket mit üblichen Kompressionsprogrammen komprimieren (wie lha, lzh, lzx, dms). Einzelne Dateien des Pakets dürfen jedoch nicht komprimiert werden (z. B. mit PowerPacker oder Imploder).

#### VERTEILUNG

Sie dürfen für Arbeit und Material nicht einen marktüblichen Preis überschreiten. Dies sind maximal 5 DM (oder ein entsprechender Wert in anderen Währungen, Steuern eingeschlossen) für Disketten und 35 DM für CD-ROMs, welche auch andere PD-Software enthalten.

Eine Verteilung auf Coverdisks oder zusammen mit kommerzieller Software bedarf in jedem Fall unserer schriftlichen Einverständnis.

Wir erlauben ausdrücklich die Verteilung über AmiNet, Meeting Pearls und Fred Fish.

#### HAFTUNGSEINSCHRÄNKUNG

Sie benutzen das Programm, wie es ist, mit allen Fehlern und Mängeln, und auf Ihr eigenes Risiko! Wir garantieren nicht dafür, daß die Software einen bestimmten Zweck erfüllt. Diese Software kann finanziellen Schaden anrichten oder Personen verletzen. In keinem Fall können die Autoren für Schäden und Mangelfolgeschäden haftbar gemacht werden, solange kein Vorsatz nachgewiesen werden kann.

#### NUTZUNGSEINSCHRÄNKUNG

Sie dürfen diese Software und ihre Ergebnisse nicht verwenden

- \* für faschistische oder militärische Zwecke
- \* wenn Sie mit dieser Copyright-Erklärung nicht einverstanden sind

In diesem Fall müssen Sie die Software und alle dazugehörigen Dateien löschen.

---

## INHALT DES PAKETS

Das Scout-Paket ist nur vollständig, wenn diese Dateien vorhanden sind:

```
Scout/arexx/activatetask.scout
Scout/arexx/arexx.readme
Scout/arexx/arexx.readme.info
Scout/arexx/breaktask.scout
Scout/arexx/freezetask.scout
Scout/arexx/poptofront.scout
Scout/arexx/settaskpri.scout
Scout/arexx/startup.scout
Scout/arexx.info
Scout/help/deutsch/scout-39.guide
Scout/help/deutsch/scout-39.guide.info
Scout/help/deutsch/scout.doc
Scout/help/deutsch/scout.doc.info
Scout/help/deutsch/scout.dvi
Scout/help/deutsch/scout.guide
Scout/help/deutsch/scout.guide.info
Scout/help/deutsch.info
Scout/help/english/scout-39.guide
Scout/help/english/scout-39.guide.info
Scout/help/english/scout.doc
Scout/help/english/scout.doc.info
Scout/help/english/scout.dvi
Scout/help/english/scout.guide
Scout/help/english/scout.guide.info
Scout/help/english.info
Scout/help.info
Scout/icons/icons.readme
Scout/icons/icons.readme.info
Scout/icons/Scout.info
Scout/icons/ScoutDock
Scout/icons/ScoutDock.info
Scout/icons.info
Scout/libs/identify.library
Scout/libs/identify.readme
Scout/libs/identify.readme.info
Scout/libs.info
Scout/Product-Info
Scout/Scout
Scout/Scout.info
Scout/scout.history
Scout/Scout.history.info
Scout/Scout.readme
Scout/Scout.readme.info
Scout.info
```

## MARKENZEICHEN

Die Copyright und Markenzeichen aller Hersteller und Produkte werden von ihren Eigentümern gehalten.

## 1.4 scout-39.guide/System Requirements

---

## Systemanforderungen

=====

Scout benötigt mindestens die Kickstart Version 2.04. Die identify.library V8 wird dringend empfohlen, da Scout ansonsten in der Funktionalität eingeschränkt ist. Siehe auch

Identify

.

Möchten Sie das Programm mit der grafischen Benutzungsoberfläche benutzen, dann müssen Sie die MUI-Version 2.1 oder eine höhere Version von MUI installieren. Siehe auch

MUI und wo man es bekommt

.

Um die Netz-Funktionen von Scout benutzen zu können, sollten Sie mindestens die AmiTCP-Version 4.0 installiert haben. Siehe auch

AmiTCP und wo man es bekommt

.

## 1.5 scout-39.guide/MUI

MUI - MagicUserInterface

=====

(C) Copyright 1992-97 Stefan Stuntz

MUI ist ein System zum Erzeugen und Unterstützen von grafischen Benutzungsoberflächen. Mit der Hilfe eines Konfigurationsprogrammes bekommt der Benutzer einer MUI-Applikation die Möglichkeit das Aussehen dieser Applikation seinem Geschmack anzupassen.

MUI wird als Shareware vertrieben. Um ein vollständiges Programmpaket zu bekommen, das viele Beispiele und mehr Informationen über die Registrierung beinhaltet, sollten Sie auf lokalen Bulletin Boards oder Public Domain Disketten nach einem File namens muiXXusr.lha Ausschau halten (XX steht für die letzte Versionsnummer).

Sie können sich auch direkt registrieren lassen, indem Sie 30.- DM oder 20.- US\$ an die folgende Adresse schicken:

Stefan Stuntz  
Eduard-Spranger-Straße 7  
80935 München  
GERMANY

## 1.6 scout-39.guide/Identify

Identify  
=====

Copyright (C) 1996-97 Richard Körber

Identify ist eine Shared Library, die Erweiterungskarten, Guru-Codes und Library-Funktionen erkennt sowie das System identifiziert.

Identify ist FreeWare. Ein vollständiges Programmpaket finden Sie im AmiNet (util/libs/Identify.lha) oder auf der Homepage des Autors: <http://www.is-koeln.de/einwohner/shred/>. Anfragen richten Sie bitte an [shred@chessy.aworld.de](mailto:shred@chessy.aworld.de).

## 1.7 scout-39.guide/AmiTCP

AmiTCP  
=====

AmiTCP ist ein TCP/IP Protokoll-Stack für den Amiga. Die Demoversion 4.0 (oder neuer) sollte in jeder größeren Public-Domain-Sammlung oder auf dem AmiNet erhältlich sein. Fragen Sie den Amiga-Händler Ihres Vertrauens. =:^)

## 1.8 scout-39.guide/Installation

Installation  
=====

Für die Installation von Scout reicht es aus, nur das Programm scout selbst in ein Verzeichnis Ihrer Wahl zu kopieren, und die identify.library in libs: abzulegen. Danach können Sie es sofort starten.

## 1.9 scout-39.guide/Using Scout

Wie wird Scout benutzt?

\*\*\*\*\*

In diesem Kapitel wird die Benutzung von Scout über die grafische Benutzungsoberfläche beschrieben. Diese grafische Benutzungsoberfläche wurde mit MUI realisiert, das für die grafische Benutzung von Scout auch im System vorhanden sein muß. Siehe auch

Was ist MUI und wo bekommt man es?

.

Möchten Sie -- aus welchem Grund auch immer -- MUI nicht verwenden,  
dann sollten Sie sich den  
Scout ohne MUI  
anschauen.

Wird das Programm gestartet, so erscheint das Hauptfenster mit  
folgenden Gadgets:

- Libraries
- Devices
- Resources
- Semaphores
- Allocations
- Tasks
- Ports
- Commodities
- Interrupts
- Vectors
- ScreenMode
- Windows
- Fonts
- BoopsiClasses
- InputHandlers
- Assigns
- Locks
- Mounted Devs
- Resident Cmds
- Timer
- System
- Expansions
- Memory
- LowMemory
- Residents

---

Jedes dieser oben dargestellten Gadgets steht für eine bestimmte Art von für das Betriebssystem notwendigen Strukturen. ←

Betätigen Sie eines dieser Gadgets, dann wird ein weiteres Fenster geöffnet, welches die jeweils dazugehörige Liste von Strukturen beinhaltet.

Beispiel: Betätigen Sie das Tasks-Gadget, so wird ein Fenster mit der aktuellen Task-Liste des Systems geöffnet.

Diese ganzen Funktionen können auch jeweils über das Menu und durch eine Taste aufgerufen werden, die durch das unterstrichene Zeichen auf jedem Gadget bestimmt wird.

Mit diesem Programm können Sie auf viele dieser Strukturen bestimmte Aktionen ausführen lassen. Sollten Sie so etwas in Betracht ziehen, dann sollten Sie sich bewußt sein, was Sie tun.

Achtung: Unsachgemäße Manipulation der System-Strukturen kann zum Absturz des Systems führen. In schweren Fällen kann dies einen Datenverlust zur Folge haben.

Hinweis: Da es für die Anleitung eines solchen Programmes zu aufwendig wäre, die angegebenen Strukturen bis ins letzte Detail zu erklären, wundern Sie sich bitte nicht, daß einige Detail-Informationen fehlen.

Da über diese Dinge schon Bücher über Bücher geschrieben wurden, verweise ich an dieser Stelle auf die dafür vorgesehene Fachliteratur!

## 1.10 scout-39.guide/Allocations

Allocations  
=====

Dieses Fenster informiert Sie darüber, welche Hardware-Resource an wen vergeben wurde.

CIA  
---

Der Amiga besitzt zwei CIAs zur Steuerung der Hardware, der Tastatur und der Druckerschnittstelle. Außerdem besitzt er mehrere Timer. Es wird angezeigt, welcher Teil der CIA noch nicht belegt wurde bzw. welches Programm ihn belegt hält.

Timer A  
Timer B

Dies sind zwei 16bit-Timer, welche gegebenenfalls auch zu einem 32bit-Timer gekoppelt werden können.

Alarm

Diese Resource informiert, wenn ein dritter Timer einen bestimmten

Zählerstand erreicht hat.

#### Serial

Dies ist eine einfache serielle Schnittstelle. Üblicherweise wird die der CIA-A zur Kommunikation mit der Tastatur verwendet, die der CIA-B ist in aller Regel nicht belegt.

#### Flag

Dies ist ein spezieller Steuereingang. Bei der CIA-A ist diese Leitung mit der Index-Leitung der Diskettenlaufwerke verbunden.

Anmerkung: in neueren DraCos befinden sich keine CIAs mehr. Diese Hardware-Ressourcen werden daher emuliert oder stehen gar nicht erst zur Verfügung.

#### Ports

-----

Dies sind die Ressourcen für die interne parallele und serielle Schnittstelle.

#### Serial Port

Dies sind die reinen Datenübertragungsregister (Senden wie Empfangen).

#### Serial Control

Dies sind die seriellen Steuerleitungen wie zum Beispiel Carrier Detect. Die Datenübertragungsleitungen gehören nicht hierzu.

#### Parallel Port

Dies sind die Datenleitungen des Parallelports.

#### Parallel Control

Dies sind die Steuerleitungen des Parallelports, zum Beispiel Busy oder Paper Out.

#### Aktionen

-----

#### Update

Betätigen Sie dieses Gadget, dann werden die Daten erneut eingelesen.

#### Print

Mit Hilfe dieser Funktion können Sie diese Angaben zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### Exit

Das Allocations-Fenster wird geschlossen.

## 1.11 scout-39.guide/Assigns

#### Assigns

=====

---

Ein Assign weist einem Verzeichnis einen logischen Namen zu.

Wenn Sie zum Beispiel einem Verzeichnis DH0:Daten/Dokumente den logischen Namen `Texte:` zuweisen, dann können Sie auf eine Datei `Dateiname`, die sich in diesem Verzeichnis befindet, auch durch die Angabe von `'Texte:Dateiname'` zugreifen.

Spalteneinträge  
-----

Address

An dieser Adresse beginnt die Struktur eines Assign-Eintrages.

Name

Logischer Name eines Verzeichnisses oder Gerätes

Path

Hier steht der Pfad des Verzeichnisses.

Aktionen  
-----

Update

Betätigen Sie dieses Gadget, dann wird die Liste erneut eingelesen.

Print

Mit Hilfe dieser Funktion können Sie die Liste der Assigns zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Remove

Mit dieser Funktion wird der ausgewählte Assign-Eintrag aus dem System entfernt.

Exit

Das Assigns-Fenster wird geschlossen.

## 1.12 scout-39.guide/BoopsiClasses

BoopsiClasses  
=====

BOOPSI-Klassen sind Objektklassen des Amiga. Die in dieser Liste dargestellten Klassen sind alle von Intuition öffentlich zur Verfügung gestellt.

Spalteneinträge  
-----

Address

An dieser Adresse beginnt die Struktur `IClass`, die sämtliche Daten für die Klasse enthält.

---

**Objects**

Zeigt die momentane Anzahl der Objekte, die mit dieser Klasse erzeugt wurden.

**Subclasses**

Zeigt die momentane Anzahl der Klassen (öffentlich wie privat), die aus dieser Klasse abgeleitet wurden.

**Superclass**

Ein Zeiger auf die IClass-Struktur der übergeordneten Klasse.

**Dispatcher**

Ein Zeiger auf den Dispatcher-Programmcode, der die Methoden der Klasse realisiert.

**Name**

Name der Klasse.

**Aktionen**

-----

**Update**

Betätigen Sie dieses Gadget, dann wird die Liste erneut eingelesen.

**Print**

Mit Hilfe dieser Funktion können Sie diese Liste zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

**Remove**

Mit dieser Funktion wird die ausgewählte Klasse aus dem System entfernt. Eine Entfernung ist nur möglich, wenn weder Objekte noch Unterklassen der Klasse existieren.

**More**

Ein Fenster mit weiteren Informationen wird geöffnet.

**Exit**

Das Fenster wird geschlossen.

## 1.13 scout-39.guide/Commodities

**Commodities**

=====

Commodities sind kleine Hilfsprogramme, die meistens auf den Eingabestrom reagieren oder ihn verändern.

Beispiele für Commodities finden Sie im Tools-Verzeichnis Ihrer Workbench.

**Spalteneinträge**

-----

---

**Address**

An dieser Adresse beginnt die Struktur CxObj, die sämtliche Daten für das Commodity enthält.

**ln\_Type**

Der Typ der Struktur. Sie ist in aller Regel vom Typ Broker.

**ln\_Pri**

Priorität des Commodity-Brokers. In der Reihenfolge der Prioritäten wird der Eingabestrom durchgereicht.

**Flags**

Flags, die den Broker näher beschreiben.

**Port**

Adresse des MessagePort, an den alle Nachrichten des Brokers gesendet werden.

**Name**

Name des Commodities.

**Aktionen**

-----

**Appear****Disappear**

Fordert das angewählte Commodity auf, seine Oberfläche zu öffnen bzw. zu schließen. Nicht alle Commodities bieten allerdings eine Oberfläche an.

**Enable****Disable**

Hiermit wird das Commodity aktiviert bzw. deaktiviert.

**Kill**

Das Commodity wird aufgefordert, sich zu beenden. Im Gegensatz zu Remove entfernt sich das Commodity dann selbst.

**ListChg****Unique**

Dem Commodity wird mitgeteilt, daß sich die Liste geändert hat bzw. daß ein weiteres Commodity mit gleichem Namen dazukommen wollte. Dies ist eigentlich nur für Programmierer zum Testen ihrer Commodities sinnvoll, sofern ihre Commodities diese Befehle unterstützen.

**Update**

Betätigen Sie dieses Gadget, dann wird die Liste erneut eingelesen.

**Print**

Mit Hilfe dieser Funktion können Sie diese Liste zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

**Priority**

Ändert die Priorität des Commodities.

---

**Remove**

Mit dieser Funktion wird der ausgewählte Commodity-Eintrag aus dem System entfernt. Bitte versuchen Sie vorher mit Kill eine sanfte Entfernung. Vielleicht geht das Commodity ja freiwillig! =;^)

**More**

Ein Fenster mit weiteren Informationen wird geöffnet.

**Exit**

Das Fenster wird geschlossen.

## 1.14 scout-39.guide/Devices

### Devices

=====

Ein Device, das sich in dieser Liste befindet, ist -- wie auch eine Library (siehe

#### Libraries

) -- eine Ansammlung von Funktionen bzw.

Routinen, denen bestimmte Aufgaben zugeordnet wurden.

Das trackdisk.device zum Beispiel beinhaltet Funktionen für die Handhabung von Disketten bzw. der Laufwerke.

### Spalteneinträge

-----

**Address**

Adresse der Device-Struktur

**ln\_Name**

Name eines Devices

**ln\_Pri**

Priorität eines Devices

**OpenC**

Zähler, der angibt, wie oft das Device geöffnet wurde.

**RPC**

RPC steht für RAM Pointer Count und gibt an, wieviele Sprungadressen des Devices ins RAM zeigen. So eine ins RAM zeigende Einsprungadresse weist auf ein Programm (z.B. den SetPatch-Befehl) hin, welches die alte Funktion verbessern bzw. erneuern möchte, indem es einfach die Sprungadresse der Funktion durch die Adresse einer eigenen Funktion ersetzt.

Viele Viren hängen sich auf diese Weise ins System. Diese Tatsache soll Sie aber jetzt nicht in Panik versetzen, da es sich in den meisten Fällen um kleine Patch-Programme -- wie den SetPatch-Befehl von Commodore -- handelt.

Sollten alle Sprungadressen eines Devices ins RAM zeigen, dann hat

es seinen Programmcode im RAM stehen. Ein solcher RPC-Eintrag besteht aus drei Sternen, da es in dem Fall unwichtig ist, wieviele Sprungadressen ins RAM zeigen.

#### In\_Type

Typ dieser Struktur (Hier sollte normalerweise device stehen.)

#### Aktionen

-----

#### Update

Die Device-Liste wird erneut ausgelesen.

#### Print

Mit Hilfe dieser Funktion können Sie die Liste der Devices zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### Remove

Mit dieser Funktion wird das ausgewählte Device entfernt. Voraussetzung hierfür ist allerdings, daß es von keinem Programm mehr benutzt wird bzw. der OpenC gleich Null ist.

#### Priority

Die Priorität des Devices kann hier von Ihnen verändert werden. Hierzu erscheint ein kleines Fenster, in dem Sie eine neue Priorität angeben können. Durch die veränderte Priorität bekommt das Device eventuell einen neuen Platz in der Device-Liste.

#### More

Ein zusätzliches Fenster wird geöffnet, in dem Sie weitere Details des selektierten Devices finden.

Sie erreichen dasselbe, indem Sie einfach einen Doppelklick auf den jeweiligen Device-Eintrag ausführen.

#### Functions

In einem Unterfenster werden sämtliche Funktionsoffsets des Devices sowie deren Einsprungadressen angezeigt. Wenn eine passende .fd-Datei existiert und der Assign FD: auf dessen Verzeichnis gesetzt ist, werden auch die Namen der Funktionen angezeigt.

#### Exit

Das Devices-Fenster wird geschlossen.

## 1.15 scout-39.guide/Expansions

### Expansions (System-Erweiterungen)

=====

WICHTIG: Alle Scout-Releases vor 2.10 sind nicht mehr kompatibel! Wenn Sie diese Liste über TCP/IP von einem Fremdrechner lesen, achten Sie bitte unbedingt darauf, daß der Zielrechner ebenfalls eine

aktuelle Version benutzt!

Dem Benutzer wird eine Liste aller System-Erweiterungen geboten, die zur Zeit dem System zur Verfügung stehen (Grafikkarten, Speichererweiterungen usw.).

Spalteneinträge

-----

Address

Die Adresse der Expansion-Struktur.

BoardAddr

Das ROM der Karte ist ab dieser Adresse im Speicher zu finden. Sollte es sich bei der Karte um eine Speichererweiterung handeln, ist hier die Anfangsadresse des konfigurierten Speichersegmentes zu finden.

Type

Nähere Informationen zur Karte. Sie werden im More-Fenster ausführlich beschrieben.

Manufacturer

Name des Kartenherstellers.

Product

Name und Klasse des Produkts.

Zusatzinformationen

-----

Wenn Sie einen Eintrag auswählen, erscheinen im Textfeld darunter noch ein paar zusätzliche Informationen:

Size

Handelt es sich bei dem Listen-Eintrag um eine Speichererweiterung, dann steht hier die Byte-Anzahl, die dem System durch diese Karte als Speicher zur Verfügung gestellt wird.

Bei normalen Karten wird hier nur die Größe des zur Karte gehörenden ROMs angegeben.

Flags

Werden im More-Fenster näher erläutert.

ID

Herstellernummer, die von Commodore vergeben wird, gefolgt von der Produktnummer, die der System-Erweiterung vom Hersteller gegeben wird.

SN

Seriennummer der Karte (Dieser Eintrag wird von den meisten Karten nicht benutzt.)

Aktionen

-----

---

#### Print

Mit Hilfe dieser Funktion können Sie die Liste der Expansions zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### More

Beim Betätigen dieses Gadgets erhalten Sie mehr Informationen über die selektierte System-Erweiterung in einem zusätzlichen Fenster.

Sie erreichen dasselbe, indem Sie einfach einen Doppelklick auf den jeweiligen Eintrag der Liste ausführen.

#### Exit

Das Expansions-Fenster wird geschlossen.

#### Unbekannte System-Erweiterungen

-----

Wenn Sie eine System-Erweiterung durch einfaches Anklicken des jeweiligen Eintrages mit der Maus selektieren, dann erhalten Sie den Namen der Herstellerfirma und die Bezeichnung der Karte in dem dafür vorgesehenen Textfeld unterhalb der Liste. Das passiert natürlich nur, sofern diese Daten der jeweilig installierten Version der identify.library bekannt sind!

Sollten diese Angaben fehlen oder nicht mit den Daten Ihrer System-Erweiterungen übereinstimmen, so möchte ich Sie bitten, mir die folgenden Daten zuzusenden, damit ich sie dem Programm beifügen bzw. sie korrigieren kann. In der nächsten Identify-Version sollten diese Angaben dann vorhanden sein.

Daten zur Erfassung einer nicht namentlich genannten Erweiterung:

1. Herstellernummer (Manufacturer)
2. Produktnummer (Product)
3. Name des Herstellers
4. Bezeichnung der Hardware
5. Funktion der Karte

Seien Sie hierbei bitte so genau wie möglich. Die Version der Erweiterung oder auch noch andere Angaben können hierbei nicht schaden.

## 1.16 scout-39.guide/Fonts

#### Fonts

=====

Alle Zeichensätze, die sich zur Zeit im System befinden bzw. von Programmen benutzt werden, sind in dieser Liste zu finden.

#### Spalteneinträge

-----

#### YSize

---

Vertikale Größe des Zeichensatzes

Count

Zähler, der angibt, von wievielen Programmen der Zeichensatz gerade benutzt wird.

Type

Steht an dieser Stelle ROMFONT, so befindet sich dieser Zeichensatz im ROM. Bei DISKFONT wurde er von Diskette bzw. Festplatte geladen.

Name

Name des Zeichensatzes

Aktionen

-----

Update

Die Liste der Zeichensätze wird aktualisiert.

Print

Mit Hilfe dieser Funktion können Sie die Liste der Fonts zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Close

Hiermit kann ein Zeichensatz geschlossen werden. Count verringert sich dann um eins.

Remove

Mit dieser Funktion kann ein Zeichensatz aus dem System (Speicher) entfernt werden, vorausgesetzt er wird von keinem Programm mehr benötigt und befindet sich nicht im ROM.

Exit

Das Fonts-Fenster wird geschlossen.

## 1.17 scout-39.guide/InputHandlers

Inputhandler

=====

Inputhandler kümmern sich um die Benutzereingaben, die im System ankommen (Tastendrucke, Mausclicks, usw.). Sie stehen wie an einem Fließband in einer Reihe und werten diese Eingaben aus. Der Inputhandler mit der höchsten Piorität bearbeitet diese Eingaben zuerst. Kann er mit den Eingaben nichts anfangen, reicht er sie in der Regel an den nächsten Inputhandler weiter.

Das System benutzt normalerweise für seinen Inputhandler die Priorität 50. Möchte also ein Inputhandler die Benutzereingaben vor dem System bekommen, braucht er eine höhere Priorität.

Spalteneinträge

-----

---

In\_Name

Name des Inpuhandlers

In\_Pri

Priorität des Inpuhandlers

is\_Data

Ab dieser Adresse sind die Daten des Inpuhandlers im Speicher zu finden.

is\_Code

Diese Adresse zeigt zum Programmcode des Inpuhandlers. Sollte diese Adresse ins RAM zeigen, so wird sie andersfarbig dargestellt. Der Inpuhandler des Betriebssystems hat seinen Programmcode im ROM.

Ein paar Viren klinken sich als Inpuhandler ins System. Bei denen zeigt dann auch die is\_Code-Adresse ins RAM. Wiederum gilt auch in einem solchen Fall: Nicht gleich die Panik bekommen, es gibt genug normale Programme, die so verfahren.

Aktionen

-----

Update

Die Liste der Inpuhandler wird auf den neuesten Stand gebracht.

Print

Mit Hilfe dieser Funktion können Sie die Liste der InputHandlers zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Remove

Ein Inpuhandler kann mit Hilfe dieser Funktion aus dem System entfernt werden. Hierbei zieht man dem System aber eventuell den Stuhl unter dem Hintern weg. Das System kann dabei leicht abstürzen!

Priority

Die Priorität des Inpuhandlers kann auf einen bestimmten Wert gesetzt werden. Wird die Priorität eines Inpuhandlers verringert, kann es passieren, daß Programme nicht mehr auf bestimmte Dinge (z.B. das Drücken einer bestimmten Taste) reagieren, da ein Inpuhandler mit einer höheren Priorität diese absorbiert.

Auch diese Liste wird vom System nach den Prioritäten sortiert. Ändern Sie also die Priorität eines Inpuhandlers, dann bekommt dieser eventuell einen neuen Platz in der Liste.

Exit

Das Fenster wird geschlossen.

---

## 1.18 scout-39.guide/Interrupts

### Interrupts

=====

Interrupts sind bestimmte Ereignisse, auf die das Betriebssystem reagieren muß. Für jeden Interrupt-Typ stehen meist sogar mehrere Interrupt-Routinen zur Verfügung. Diese Interrupt-Routinen werden in einer Liste nach Prioritäten sortiert.

Sobald also ein bestimmter Interrupt auftritt, wird das laufende Programm solange unterbrochen, bis die zum jeweiligen Interrupt gehörende Liste der Interrupt-Routinen abgearbeitet wurde.

### Spalteneinträge

-----

#### In\_Name

Diesem Text kann normalerweise entnommen werden, von welchem Programm die Interrupt-Routine installiert wurde und auch benötigt wird.

#### In\_Pri

Priorität der Interrupt-Routine

#### is\_Data

Ab dieser Adresse sind im Speicher Daten zu finden, die zur Interrupt-Routine gehören.

#### is\_Code

Der Programmcode der Interrupt-Routine ist hier zu finden. Sollte diese Adresse ins RAM zeigen, so wird sie andersfarbig dargestellt.

#### NUM

Diese Nummer beschreibt das Ereignis, bei dem die Interrupt-Routine aufgerufen wird. Eine kleine Information hierzu finden Sie im IntName-Eintrag des Interrupt-Detail-Fensters, das durch das Betätigen des More-Gadgets geöffnet wird.

Beispiel: Nummer 5 bedeutet, daß die Interrupt-Routine bei jedem neuen Bildaufbau ihres Monitors aufgerufen wird, was bei einem 50 Hz Monitor 50 mal in der Sekunde passiert. (VERTB (vertical blank interval))

### Aktionen

-----

#### Update

Die Liste der Interrupt-Routinen wird aktualisiert.

#### Print

Mit Hilfe dieser Funktion können Sie die Liste der Interrupt-Routinen zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### Remove

Mit dieser Funktion kann eine Interrupt-Routine aus der Liste entfernt werden. Sollte es sich bei der Interrupt-Routine allerdings um einen Interrupt-Handler handeln, kann Scout diese Aktionen nicht ausführen. Ist dies der Fall, dann steht in der Spalte IntType der Text Handler.

Bei den Interrupt-Handlern vom audio.device kann dieses Problem z.B. gelöst werden, indem das audio.device entfernt wird. Das passiert unter anderem durch den Aufruf von avail flush, wenn das audio.device von keinem Programm mehr benutzt wird.

More

Ein Fenster mit weiteren Informationen über den selektierten Interrupt wird geöffnet.

Exit

Betätigen Sie dieses Gadget, dann wird das Fenster geschlossen.

## 1.19 scout-39.guide/Libraries

Libraries

=====

Eine Library ist eine Ansammlung von Funktionen/Routinen (Bibliothek), denen bestimmte Aufgaben zugeordnet wurden.

Die graphics.library zum Beispiel beinhaltet Funktionen für die Grafikdarstellung.

Spalteneinträge

-----

Address

Adresse einer Library

ln\_Name

Name einer Library

ln\_Pri

Priorität einer Library

OpenC

Zähler, der angibt, wie oft die Library geöffnet wurde.

RPC

RPC steht für RAM Pointer Count und gibt an, wieviele Sprungadressen der Library ins RAM zeigen. So eine ins RAM zeigende Einsprungadresse weist auf ein Programm (z.B. den SetPatch-Befehl) hin, welches die alte Funktion verbessern bzw. erneuern möchte, indem es einfach die Sprungadresse der Funktion durch die Adresse einer eigenen Funktion ersetzt.

Viele Viren hängen sich auf diese Weise ins System. Diese Tatsache soll Sie aber jetzt nicht in Panik versetzen, da es sich

in den meisten Fällen um kleine Patch-Programme -- wie den SetPatch-Befehl von Commodore -- handelt.

Sollten alle Sprungadressen einer Library ins RAM zeigen, dann hat sie ihren Programmcode im RAM stehen. Ein solcher RPC-Eintrag besteht aus drei Sternen, da es in dem Fall unwichtig ist, wieviele Sprungadressen ins RAM zeigen.

#### In\_Type

Typ dieser Struktur (Hier sollte normalerweise library stehen.)

#### Aktionen

-----

#### Priority

Die Priorität der Library kann von Ihnen verändert werden. Hierzu erscheint ein kleines Fenster, in dem Sie die neue Priorität angeben können. Durch die veränderte Priorität bekommt die Library eventuell einen neuen Platz in der Liste.

#### Close

Um eine Library aus dem System entfernen zu können, muß sie von allen Programmen wieder geschlossen worden sein. Dies ist der Fall, wenn der OpenC-Eintrag den Wert Null hat.

Wenn Sie mit dieser Funktion eine Library schließen möchten, werden Sie gefragt, ob Sie die Library nur einmal oder gleich für alle Programme schließen möchten, die diese Library geöffnet haben.

Wählen Sie hier also all, dann wird die Library so oft geschlossen, bis der OpenC gleich Null ist.

#### Remove

Mit dieser Funktion wird die selektierte Library entfernt. Voraussetzung hierfür ist allerdings, daß sie von keinem Programm mehr benutzt wird bzw. der OpenC gleich Null ist.

Einige Libraries lassen sich nicht mehr ohne einen Reset aus dem System entfernen. Es ist also nicht unbedingt verwunderlich, wenn Scout es einmal nicht schaffen sollte, eine Library zu entfernen!

#### Functions

In einem Unterfenster werden sämtliche Funktionsoffsets der Library sowie deren Einsprungadressen angezeigt. Wenn eine passende .fd-Datei existiert und der Assign FD: auf dessen Verzeichnis gesetzt ist, werden auch die Namen der Funktionen angezeigt.

#### Update

Die Library-Liste wird erneuert.

#### Print

Mit Hilfe dieser Funktion können Sie die Liste der Libraries zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### More

---

Ein Fenster mit weiteren Informationen zur Library wird geöffnet.

Exit

Das Libraries-Fenster wird geschlossen.

## 1.20 scout-39.guide/Locks

Locks

=====

Ein Lock symbolisiert den Zugriff eines Programmes auf eine Datei oder ein Verzeichnis. Auf diese Weise wird z.B. verhindert, daß eine Datei gelöscht wird, während irgendein anderes Programm noch auf die sich in der Datei befindenden Daten zugreift.

Bei etwas umfangreicheren Systemen kann der Aufbau der Liste etwas länger dauern! Mein eigenes System hat z.B. im Durchschnitt ca. 500 Lockeinträge, was gemessen an anderen Systemen noch nicht allzu viel ist. =:^)

Spalteneinträge

-----

Access

Hier wird die Zugriffsart des Lock-Zugriffes angegeben. Dies kann ein Lese- (READ) oder ein Schreibzugriff (WRITE) sein. Sollte hier OWN stehen, dann handelt es sich nur um einen Lock, der zum Aufbau dieser Liste von Scout angefordert wurde.

Path

Pfad der Datei oder des Verzeichnisses

Aktionen

-----

Update

Die Liste der Locks wird aktualisiert.

Print

Mit Hilfe dieser Funktion können Sie die Liste der Locks zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Remove

Ein Lock wird mittels der UnLock()-Funktion der dos.library wieder freigegeben.

Pattern

Geben Sie hier ein Namensmuster an, so werden nur die Locks angezeigt, deren Pfad mit dem Namensmuster übereinstimmt.

Exit

Das Fenster wird geschlossen.

---

## 1.21 scout-39.guide/LowMemory

LowMemory  
=====

In dieser Liste werden alle dem System bekannten LowMemory-Handler angezeigt.

Diese Handler werden der Reihe nach aufgerufen, wenn eine Speicherbelegung wegen mangelnden Speichers nicht ausgeführt werden konnte. Die Handler versuchen, unbenutzten Speicher wieder freizugeben, damit die Speicheranforderung eventuell doch noch ausgeführt werden kann.

'ramlib' ist ein LowMemory-Handler, der unbenötigte Libraries und Devices wieder aus dem Speicher entfernt. Er ist stets vorhanden.

Hinweis: LowMemory-Handler existieren erst seit AmigaOS 3.0. Unter älteren Versionen ist diese Liste stets leer.

Spalteneinträge  
-----

Address

Adresse der Struktur, die Daten über den LowMemory-Handler beinhaltet.

In\_Name

Name des Handlers.

In\_Type

Typ des Handlers.

In\_Pri

Priorität des Handlers. In dieser Reihenfolge werden die Handler der Reihe nach aufgerufen. Dazwischen wird stets geprüft, ob die Speicheranforderung bereits erfüllt werden kann.

is\_Data

Ein Zeiger auf die Datenstruktur, die der jeweilige Handler benutzt.

is\_Code

Ein Zeiger auf den LowMemory-Handler selbst.

Aktionen  
-----

Update

Die Liste wird aktualisiert.

Print

Mit Hilfe dieser Funktion können Sie die Liste zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Remove

---

Der LowMemory-Handler wird aus dem Speicher entfernt.

#### Priority

Hiermit bestimmen Sie die Priorität und damit die Reihenfolge, in der die Memory-Handler abgearbeitet werden.

#### Exit

Das LowMemory-Fenster wird geschlossen.

## 1.22 scout-39.guide/Memory

### Memory (Speichersegmente)

=====

Die Einträge dieser Liste stellen die Speichersegmente Ihres Rechners dar. Sie finden dort mindestens den Eintrag Ihres Grafik-Speichers (CHIP-MEMORY), der fest in Ihren Rechner eingebaut ist.

#### Spalteneinträge

-----

#### ln\_Name

Name des Speichersegmentes (z.B. chip memory)

#### ln\_Pri

Priorität des Speichersegmentes

#### mh\_Lower

Anfangsadresse des Speichersegmentes

#### mh\_Upper

Endadresse des Speichersegmentes

#### Aktionen

-----

#### Print

Mit Hilfe dieser Funktion können Sie die Liste der Speichersegmente zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### Priority

Mit dieser Funktion können Sie bestimmen, welches Speichersegment bevorzugt vom System und den anderen Programmen benutzt werden soll, indem Sie diesem eine höhere Priorität geben als den anderen Speichersegmenten.

Ausnahme: Wird der Typ des Speichers direkt bei der Anforderung eines Programmes angegeben, wird das erste Speichersegment benutzt, das die Anforderungskriterien erfüllt.

#### More

Ein neues Fenster wird geöffnet. Dieses Fenster enthält weitere Daten zum selektierten Speichersegment.

Exit

Das Memory-Fenster wird geschlossen.

## 1.23 scout-39.guide/Mounted Devs

Mounted Devices

=====

In dieser Liste finden Sie alle Ihre ansprechbaren Geräte (Laufwerke, Festplatten usw.).

Spalteneinträge

-----

Name

Name des Gerätes

Unit

Kennziffer des Gerätes (Bei DF2: steht hier z.B. normalerweise eine Zwei.)

Heads

Anzahl der vorhandenen Lese- bzw. Schreib-Köpfe

Cyl

Anzahl der Zylinder

State

Zustand eines Gerätes, der z.B. angibt, ob eine Diskette im Laufwerk liegt oder ob die Diskette unlesbar ist.

DiskType

Typ der Diskette (z.B. OFS (OldFileSystem), FFS (FastFileSystem), ...)

Handler or Device

Hier wird angegeben, welcher Handler oder welches Device sich um den Zugriff auf das jeweilige Gerät kümmert.

Beim Laufwerk DF0: wäre es z.B. in der Regel das trackdisk.device. Um also direkt auf die Sektoren von DF0: schreiben zu können, müßten Sie das trackdisk.device benutzen.

Aktionen

-----

Update

Die Liste wird erneut eingelesen.

Print

Mit Hilfe dieser Funktion können Sie die Liste der Geräte zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

More

Ein weiteres Fenster mit mehr Informationen zum ausgewählten Gerät wird geöffnet.

Exit

Das Fenster wird geschlossen.

## 1.24 scout-39.guide/Ports

Ports

=====

Ports dienen der Kommunikation von Programmen. Dem Port eines Programmes können Mitteilungen gesendet werden, auf die das Programm reagieren soll.

Spalteneinträge

-----

Address

An dieser Adresse ist die Port-Struktur zu finden.

ln\_Name

Name des Ports

ln\_Pri

Priorität des Ports

mp\_SigTask

Name des Tasks, der für diesen Port zuständig ist.

Aktionen

-----

Update

Die Liste der Ports wird aktualisiert.

Print

Mit Hilfe dieser Funktion können Sie die Liste der Ports zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Remove

Der Port wird aus dem System entfernt.

Priority

Mit Hilfe dieser Funktion kann die Priorität des Ports verändert werden.

More

Ein neues Fenster wird geöffnet. Dieses Fenster enthält weitere Daten zum selektierten Port.

Exit

Das Ports-Fenster wird geschlossen.

---

## 1.25 scout-39.guide/Resident Cmds

Resident Commands (Residente Befehle)

=====

Alle Kommandos, die durch den Shell-Befehl resident resident gemacht wurden, und die Befehle, die schon im ROM enthalten sind, werden hier angezeigt.

Dabei werden auch die Positionen und die Größen aller Hunks der jeweiligen Befehle aufgelistet.

Die hier behandelten 'residenten Befehle' haben nichts mit den im nächsten Abschnitt beschriebenen 'residenten Strukturen' zu tun.

Spalteneinträge

-----

Name

Name des Befehls

UseCount

Zähler, der angibt, wieviele Instanzen des Befehls zur Zeit des Listenaufbaus im System aktiv sind.

Lower

Startadresse eines Hunks im Speicher

Upper

Endadresse eines Hunks im Speicher

Size

Größe des Hunks (Upper - Lower - 8 Bytes Overhead)

Aktionen

-----

Update

Die Liste der residenten Befehle wird erneut eingelesen.

Print

Mit Hilfe dieser Funktion können Sie die Liste der residenten Befehle zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Remove

Mit dieser Funktion wird der ausgewählte residente Befehl aus der Liste entfernt. Voraussetzung hierfür ist allerdings, daß er nicht mehr benutzt wird bzw. der UseCount gleich Null ist.

Exit

Das Fenster wird geschlossen.

---

## 1.26 scout-39.guide/Residents

Residents (Residente Strukturen)

=====

Residente Strukturen (Residents) sind Code- bzw. Daten-Segmente (wie zum Beispiel Libraries), die einen Reset überstehen. Sie sind reset-fest.

Die hier behandelten 'residenten Strukturen' haben nichts mit den im vorigen Abschnitt beschriebenen 'residenten Befehlen' zu tun.

Ein Programmierer hat nun die Möglichkeit sein Programm reset-fest zu machen, indem er unter anderem eine Resident-Struktur initialisiert und diese über die Kick-Vektoren (siehe

Vectors

), die sich in der

ExecBase-Struktur (Basis der exec.library) befinden, ins System einklinkt.

Diese residenten Strukturen liegen demnach im RAM und ihre Adressen werden andersfarbig dargestellt, um sie von den anderen residenten Strukturen abzuheben. Die residenten Strukturen, die über die Kick-Vektoren ins System gekommen sind, werden, sofern überhaupt solche residenten Strukturen vorhanden sind, am oberen Ende der Liste eingefügt.

Sollten Sie hier eine residente Struktur finden, die ins RAM zeigt, dann ist Vorsicht geboten. Schauen Sie sich ihren Namen an, und wenn Sie nicht ganz sicher wissen, worum es sich handelt, sollten Sie lieber einmal den Virenkiller Ihres Vertrauens das System überprüfen lassen.

Viele Viren machen sich auf diese Weise reset-fest!

Spalteneinträge

-----

Address

An dieser Adresse ist die residente Struktur zu finden.

ln\_Name

Name der residenten Struktur

rt\_Pri

Priorität der residenten Struktur

rt\_IdString

Identifikationstext der residenten Struktur

Aktionen

-----

Update

Die Liste der residenten Strukturen wird aktualisiert.

#### Print

Mit Hilfe dieser Funktion können Sie die Liste der Residents zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### More

Ein neues Fenster mit mehr Informationen über die Resident-Struktur wird geöffnet.

#### Exit

Das Residents-Fenster wird geschlossen.

## 1.27 scout-39.guide/Resources

### Resources (Ressourcen)

=====

Eine Ressource ist -- wie auch eine Library (siehe Libraries) und ein Device (siehe Devices) -- eine Ansammlung von Funktionen bzw. Routinen, denen bestimmte Aufgaben zugeordnet wurden.

#### Spalteneinträge

-----

#### Address

Adresse der Ressource

#### ln\_Name

Name der Ressource

#### ln\_Pri

Priorität der Ressource

#### OpenC

Zähler, der angibt, wie oft die Ressource geöffnet wurde.

#### RPC

RPC steht für RAM Pointer Count und gibt an, wieviele Sprungadressen der Ressource ins RAM zeigen. So eine ins RAM zeigende Einsprungadresse weist auf ein Programm hin (wie z.B. den SetPatch-Befehl), welches die 'alte' Funktion verbessern bzw. erneuern möchte, indem es einfach die Sprungadresse der Funktion durch die Adresse einer eigenen Funktion ersetzt.

Sollten alle Sprungadressen einer Ressource ins RAM zeigen, dann hat sie ihren Programmcode im RAM stehen. Ein solcher RPC-Eintrag besteht aus drei Sternen, da es in dem Fall unwichtig ist, wieviele Sprungadressen ins RAM zeigen.

### In\_Type

Typ der Struktur (Hier sollte normalerweise resource stehen.)

### Aktionen

-----

### Update

Die Ressource-Liste wird neu eingelesen.

### Print

Mit Hilfe dieser Funktion können Sie die Liste der Resources zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

### Remove

Mit dieser Funktion wird die gewählte Ressource entfernt. Voraussetzung hierfür ist allerdings, daß sie von keinem Programm mehr benutzt wird bzw. der OpenC gleich Null ist.

### Priority

Die Priorität der Ressource kann von Ihnen verändert werden. Hierzu erscheint ein kleines Fenster, in dem Sie die neue Priorität angeben können.

### More

Wird dieses Gadget betätigt, dann erscheint ein zusätzliches Fenster mit weiteren Daten zur selektierten Ressource.

### Functions

In einem Unterfenster werden sämtliche Funktionsoffsets der Resource sowie deren Einsprungadressen angezeigt. Wenn eine passende .fd-Datei existiert und der Assign FD: auf dessen Verzeichnis gesetzt ist, werden auch die Namen der Funktionen angezeigt. Einige Resources bieten keine Funktionen an.

### Exit

Das Residents-Fenster wird geschlossen.

Beachte: Sollte bei OpenC und/oder RPC ein Strich stehen, so besitzt die Ressource keine typische Library-Struktur (Hintereinanderreihung von Sprungbefehlen und deren Sprungadressen). Das passiert z.B. beim Eintrag der FileSystem.resource.

## 1.28 scout-39.guide/ScreenMode

### ScreenMode

=====

Durch ScreenModes werden die einzelnen Bildschirm-Darstellungsarten definiert, die das System darstellen kann.

Sie haben sicherlich schon einmal einen Bildschirm-Modus ausgewählt. In dieser Liste finden sie sämtliche, die zur Verfügung stehen. Viele

---

Programme filtern einen Teil davon heraus, weil sie für ihre Zwecke nicht geeignet sind.

#### Spalteneinträge

-----

#### ModeID

Eine einmalige Zahl, mit der der Bildschirm-Modus verwaltet wird.

#### Width

Standard-Breite dieses Bildschirmmodus in Pixeln.

#### Height

Standard-Höhe dieses Bildschirmmodus in Pixeln.

#### Depth

Maximale Anzahl der Bitplanes. Damit zusammen hängen die Anzahl der Farben, die gleichzeitig dargestellt werden können.

#### ScreenMode

Name dieses Bildschirmmodus. Einige Modis liefern keinen Namen mit. Er wird dann von Scout generiert und kann von den Bezeichnungen in anderen Programmen und Requestern abweichen. Folgende Darstellungsarten werden ausgewiesen:

#### HAM

Spezieller Hold-And-Modify-Modus des Amiga-Chipsatzes. Damit können trotz 6 bzw. 8 Bitplanes tausende von Farben gleichzeitig dargestellt werden.

#### EHB

Extra-Halfbrite-Modus des Amiga-Chipsatzes. Es stehen 32 frei wählbare Farben sowie deren halbhellen Farbtöne zur Verfügung.

#### DualPF

#### DualPF2

Spezieller Darstellungsmodus des Amiga-Chipsatzes, bei dem der Bildschirm quasi aus zwei voneinander unabhängigen 'Folien' besteht, die übereinandergelegt wurden. Der Unterschied zwischen DualPF und DualPF2 ist die Reihenfolge, in der die Folien übereinanderliegen.

#### Interlace

Das Bild wird im Zeilensprungverfahren dargestellt.

#### DblScan

Das Bild wird im Zeilenverdoppelungsverfahren dargestellt (AGA und Grafikkarten).

#### Aktionen

-----

#### Update

Die Liste wird erneut eingelesen.

#### Print

---

Mit Hilfe dieser Funktion können Sie die Liste zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### More

Es werden weitere Informationen über diesen ScreenMode ausgegeben, mitunter die Größe der Overscan-Auflösungen sowie die Zeilen- und Bildwechselfrequenz. Durch eine Ungenauigkeit des Betriebssystems bedingt weichen die tatsächlichen Frequenz ein wenig von den dargestellten ab!

#### Exit

Das Fenster wird geschlossen.

## 1.29 scout-39.guide/Semaphores

### Semaphores (Semaphore)

=====

Semaphore sind normalerweise dafür da, den Zugriff auf bestimmte Geräte zu handhaben, auf die nur eine bestimmte Anzahl von Programmen zur Zeit zugreifen darf.

#### Beispiele:

1. Auf einen Drucker darf nur ein Programm zur Zeit zugreifen, da sonst die zu druckenden Texte 'gemischt' würden.
2. Wenn der SetPatch-Befehl von Commodore z.B. schon die Routinen des Betriebssystems gepatcht hat, dann soll er diese Patches beim nächsten Aufruf ja nicht nochmal ausführen. Zu diesem Zweck wird ein Semaphor eingerichtet. Der SetPatch-Befehl kann dadurch bei einem erneuten Start prüfen, ob er schon einmal ausgeführt worden ist.

#### Spalteneinträge

-----

#### In\_Name

Name des Semaphors

#### Nest

Dieser Zähler zeigt, wie oft der Owner-Task den Semaphor benutzt.

#### Queue

Hier wird angezeigt, wieviele Tasks den Semaphor besitzen möchten.

#### Owner

Hier ist der Name des Tasks zu finden, dem der Semaphor zur Zeit gehört.

#### Aktionen

-----

#### Update

Die Liste der Semaphore wird erneut eingelesen.

#### Print

Mit Hilfe dieser Funktion können Sie die Liste der Semaphore zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### Obtain

Hierdurch wird dem System vorgegaukelt, daß das Gerät, das File oder wofür der Semaphor sonst eingerichtet wurde, gerade benutzt wird. Der NestCnt-Eintrag erhöht sich hierbei um Eins.

#### Release

Sollte ein Semaphor gerade benutzt werden, so machen Sie dem System mit dieser Funktion weis, daß dem nicht mehr so ist. Ein Programm, das den Semaphor beachtet, kann so eventuell versuchen, ein weiteres Mal auf das entsprechende Gerät zuzugreifen.

#### Remove

Sofern der Semaphor nicht mehr benutzt wird, können Sie ihn anhand dieser Funktion aus dem System entfernen.

#### Exit

Das Semaphores-Fenster wird geschlossen.

### 1.30 scout-39.guide/System

#### System

=====

#### Aktionen

-----

#### Print

Mit Hilfe dieser Funktion können Sie diese Liste zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

#### Exit

Das System-Fenster wird geschlossen.

#### Einträge

-----

In dem Fenster finden Sie einige, zum Teil sehr technische Angaben über Ihren Computer. Bitte entschuldigen Sie hierbei das gewöhnliche Aussehen des Fensters, aber dadurch läßt es sich sehr bequem erweitern.

### 1.31 scout-39.guide/Tasks

#### Tasks

=====

---

In dieser Liste befinden sich alle Tasks und Prozesse. (Prozesse sind erweiterte Task-Strukturen.) Sie repräsentieren die Programme, die im Augenblick im System ablaufen bzw. auf ein Ereignis warten.

Spalteneinträge  
-----

ln\_Name

Name des Tasks

ln\_Type

Typ der Struktur (task oder process)

ln\_Pri

Priorität des Tasks

NUM

Hier steht die Nummer eines Prozesses, sofern dieser sich mit Hilfe des Befehles run abgekoppelt hat oder noch in einer Shell läuft. Ein Programm, das über die Workbench gestartet wurde, hat als NUM-Eintrag einen Strich, wie auch ein Programm, das sich selbständig von der Shell abgekoppelt hat.

State

Dieser Eintrag zeigt den Zustand eines Tasks/Prozesses an. Der eigene Prozess von Scout, der ganz oben in der Liste zu finden ist, hat dort immer run stehen, weil er immer aktiv ist, wenn er die Task-Liste ausliest. =:^)

Ein wait bedeutet hierbei, daß ein Task auf ein bestimmtes Ereignis wartet. Dies kann zum Beispiel das Betätigen eines Gadgets sein.

Sollte sich ein Task im Zustand ready befinden, dann hat er zwar gerade etwas zu tun, wurde aber von der Abarbeitung eines anderen Prozesses unterbrochen (Multitasking-Prinzip).

SigWait

Signalmaske, auf die der Task wartet. Sollte ein Task im Zustand wait sein und diese Signalmaske den Wert Null (\$00000000) haben, dann handelt es sich mit großer Wahrscheinlichkeit um einen Task, der sich 'aufgehängt' hat und vom Betriebssystem in der Schwebe gehalten wird. (suspend or reboot)

Aktionen  
-----

Print

Mit Hilfe dieser Funktion können Sie die Liste der Tasks zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Freeze

Hiermit wird ein Task eingefroren. Er befindet sich zwar dann noch in der Task-Liste, bekommt aber keine Rechenzeit mehr vom System.

Achtung: Wenn Sie versuchen Tasks einzufrieren, die für das System

lebenswichtig sind (wie z.B. der Task `input.device`), sollten Sie alle wichtigen Daten abgespeichert haben, da durch den folgenden Systemabsturz diese Daten sonst verloren sind.

#### Activate

Ein eingefrorener Task kann hiermit wieder aktiviert werden.

#### CPU

Hier finden Sie ein Textfeld und ein Cycle-Gadget. Das Textfeld gibt -- abhängig von dem Zustand des Cycle-Gadgets -- die verbrauchte CPU-Auslastung in Prozent an.

Für das Cycle-Gadgets gibt es drei Zustände:

##### off

In diesem Zustand wird die CPU-Auslastung nicht berechnet.

##### full

Wurde dieser Zustand gewählt, dann setzt Scout die verbrauchte CPU-Auslastung auf 100%, d.h. die Summe der CPU-Auslastungsprozente aller in der Liste stehenden Tasks und Prozesse ergibt immer 100%. Dies ist unabhängig von der wirklich verbrauchten Rechenzeit.

##### in %

In diesem Fall wird die wirklich verbrauchte CPU-Auslastung gemessen und in dem dafür vorgesehenen Textfeld angegeben. Dafür startet Scout den Task « Scout's cheat task », der mit der Priorität -128 die ganze nicht verbrauchte Prozessorzeit beansprucht.

#### Secs

Mit Hilfe dieses String-Gadgets können Sie bestimmen, in welchen Intervallen die CPU-Auslastung gemessen wird, sofern Sie diese Funktion beim Cycle-Gadget mittels `full` oder `in %` überhaupt ausgewählt haben. Dieses Intervall sollte nicht zu klein gewählt werden, da es zu Ungenauigkeiten kommen kann und Scout dann die meiste Rechenzeit beansprucht. Intervalle kleiner 0.5 Sekunden machen nicht viel Sinn!

#### Update

Die Liste der Tasks und Prozesse wird erneut eingelesen.

#### Remove

Ein Task wird aus der Liste entfernt. Sollten Sie sich nicht ganz sicher sein, ob Sie den Task noch einmal brauchen, dann sollten Sie lieber die Freeze-Funktion benutzen. (Siehe auch Break!)

#### Signal

Sie können beim Benutzen dieser Funktion eine Signalmaske angeben, die darauf dem ausgewählten Task geschickt wird.

#### Break

Einem Task wird ein Break-Signal gesendet. Viele Tasks reagieren auf dieses Signal und beenden sich selbst. Reagiert der Task, der mit Hilfe von Scout aus dem System entfernt werden soll, auf dieses Signal, dann sollte er normalerweise den von ihm angeforderten Speicher wieder freigeben. Wird ein Task durch die Remove-Funktion

entfernt, wird der von ihm benutzte Speicher nicht wieder freigegeben. Es bleiben dann sogenannte 'Speicherleichen' im System zurück.

#### Priority

Die Priorität eines Tasks kann hiermit verändert werden. Ein Task mit einer niedrigen Priorität bekommt erst vom System Rechenzeit zur Verfügung gestellt, wenn kein Task mit einer höheren Priorität Rechenzeit benötigt.

#### More

Ein weiteres Fenster wird geöffnet, das, je nachdem ob ein Task oder ein Prozess selektiert wurde, weitere Informationen zu dem Task oder dem Prozess beinhaltet.

#### Exit

Das Fenster mit der Task-Liste wird geschlossen.

## 1.32 scout-39.guide/Timer

### Timer

=====

Dieses Fenster enthält alle Anfragen an das timer.device.

#### Spalteneinträge

-----

#### Address

Die Adresse der IORequest-Struktur.

#### ReplyPort

Die Adresse des MessagePorts, an den der Request zurückgeschickt wird.

#### Time

Zeit, die dieser Request noch in der Liste verbringt.

#### Unit

Es gibt zwei verschiedene Meßarten: VBlank (weniger Overhead) und MicroHz (genauerer Maß).

#### Task

Der Name des anfragenden Tasks.

#### Actions

-----

#### Print

Diese Funktion erlaubt es Ihnen, die Liste an einen Drucker oder in eine ausgewählte Datei zu schicken.

#### Update

Die Liste wird aktualisiert.

---

Exit

Das Fenster wird geschlossen.

### 1.33 scout-39.guide/Vectors

Vectors (Spezielle Vektoren)

=====

Aktionen

-----

Update

Die Vektoren werden erneut ausgelesen.

Print

Mit Hilfe dieser Funktion können Sie die Liste der Vektoren zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Exit

Das Vectors-Fenster wird geschlossen.

Reset Vectors

-----

Mit Hilfe der Reset-Vektoren kann sich ein Programm reset-fest ins System einhängen. Sie haben einen Wert von Null, wenn sie nicht verbogen wurden. Benutzt ein Programm die Kick-Vektoren (KickTagPtr, KickMemPtr und KickChecksum) um sich reset-fest zu machen, dann ist es auch in der Liste der residenten Strukturen zu finden. Siehe auch

Residents

.

Auto Vector Interrupts

-----

Die sieben Auto-Vektor-Interrupts, die hier angezeigt werden, sind bei einem System mit MC68000-Prozessor von Adresse \$64 bis \$7c zu finden. Die Prozessoren MC68010 und aufwärts besitzen ein Vektor-Basis-Register (VBR), das eine Verlegung der Interrupt-Tabelle ins FAST-RAM ermöglicht. Durch diese Verlegung ins FAST-RAM wird das System etwas beschleunigt. Scout berücksichtigt das VBR bei der Darstellung dieser Vektoren, vorausgesetzt es ist vorhanden und wird benutzt.

Interrupt Vectors

-----

Die hier angezeigten 16 Interrupt-Vektoren (IntVecs) befinden sich in der ExecBase-Struktur (der Basisstruktur der exec.library). Welche Aufgabe sie haben bzw. wie das Zusammenspiel der Auto-Vektor-Interrupts, der Interrupt-Vektoren und der Interrupt-Handler bzw. Interrupt-Server (siehe

Interrupts  
)

funktioniert, entnehmen Sie bitte der Fachliteratur.

## 1.34 scout-39.guide/Windows

Windows (Fenster)  
=====

In dieser Liste werden alle Screens mit den auf ihnen befindlichen Fenstern angezeigt. Screens werden andersfarbig dargestellt, damit sie sich besser von den Fenstern unterscheiden.

Spalteneinträge  
-----

Pos(x,y)  
    Horizontale (X) und vertikale (Y) Position des Screens/Fensters

Size(x,y)  
    Horizontale (X) und vertikale (Y) Größe des Screens/Fensters

Title  
    Titel des Screens/Fensters

Aktionen  
-----

Update  
    Die Liste wird erneut eingelesen.

Print  
    Mit Hilfe dieser Funktion können Sie die Liste der Fenster zum Drucker schicken oder in eine Datei Ihrer Wahl ausgeben lassen.

Close  
    Ihnen wird hiermit die Möglichkeit gegeben, Fenster und Screens zu schließen. Ein Screen wird dann mit all den Fenstern geschlossen, die sich auf ihm befinden.

ToFront  
    Das von selektierte Element der Windows- und Screensliste wird in den Vordergrund geholt.

More  
    Je nachdem, ob ein Screen oder ein Fenster in der Liste selektiert wurde, wird ein weiteres Fenster geöffnet, das weitere Daten zum Screen oder zum Fenster enthält.

Exit  
    Das Windows-Fenster wird geschlossen.

---

## 1.35 scout-39.guide/Scout and AmiTCP

Scout und AmiTCP

=====

Dieser Abschnitt soll Ihnen kurz erläutern, was Sie machen müssen, um Ihren Rechner durch Scout und AmiTCP von einem anderen Rechner aus beeinflussen zu können.

Es werden hier bestimmte Kenntnisse zu AmiTCP vorausgesetzt. Wenn Sie mit diesem Thema absolut nichts anfangen können, dann können Sie der Anleitung von AmiTCP entnehmen, was es alles damit auf sich hat!

(Siehe auch

AmiTCP und wo man es bekommt  
.)

Das Programm Scout dient unter AmiTCP als Client und als Server. Demnach brauchen Sie also neben der AmiTCP-Installation kein zusätzliches Programm, um Scout zusammen mit AmiTCP benutzen zu können.

Möchten Sie Ihren Rechner einem anderen System via Scout zugänglich machen, dann müssen Sie die nun folgenden zwei Schritte ausführen:

1. Fügen Sie dem File AmiTCP:db/services die Zeile scout 6543/tcp hinzu.
2. Jetzt fügen Sie bitte dem File AmiTCP:db/inetd.conf die Zeile scout stream tcp nowait root dh0:scout hinzu. Hierbei ist zu beachten, daß unter dem Pfad am Ende der Zeile wirklich das Programm Scout zu finden ist. Korrigieren Sie ggf. diesen Pfad in der Textzeile!

Das war's! Wenn Sie nun AmiTCP starten, dann ist Ihr Rechner prinzipiell von anderen System aus über Scout unter Verwendung der Optionen HOST, USER und PASSWORD erreichbar.

Beispiel: Wenn ich die Systemstrukturen meines Rechners von einem anderen System aus warten möchte. Dann müßte ich (natürlich mit einem anderen Passwort) Scout wie folgt aufrufen:

```
1> scout HOST crash.north.de USER atte PASSWORD secret
```

Wird die Option PASSWORD weggelassen, dann werden Sie nachträglich aufgefordert, das Passwort in der Shell einzugeben. Diese Variante ist sicherer, falls Sie nicht allein sind und Ihr Passwort nicht preisgeben möchten, da das Passwort, das Sie in der Shell eingeben, nicht dargestellt wird.

Auch die Option USER kann weggelassen werden. In diesem Fall nimmt AmiTCP an, daß derselbe Username verwendet werden soll, unter dem Sie sich derzeit in Ihrem System aufhalten.

Auch bei der Verwendung von AmiTCP sind Sie nicht daran gebunden MUI installiert zu haben. Alle Shell-Befehle (siehe

ARexx- und Shell-Befehle  
 ) können auch zusammen mit AmiTCP verwendet

werden.

Beispiel: Möchte ich z.B. die aktuelle Taskliste meines Rechners von einem anderen System aus ausgeben lassen. Dann müßte ich (natürlich wieder mit einem anderen Passwort) Scout wie folgt aufrufen:

```
1> scout HOST crash.north.de USER atte PASSWORD secret Tasks
```

Um die Angabe des korrekten Passwortes kommen Sie, wie jeder andere Benutzer, in 'keinem' Fall herum. Jeder, der Ihr System durch Scout beeinflussen möchte, muß ein Login auf Ihrem Rechner haben und sich korrekt identifizieren. Desweiteren gibt es bei AmiTCP durch einen Eintrag in der Datei AmiTCP:db/inet.access) auch die Möglichkeit, bestimmte Services für beliebige Systeme zu sperren. Wenn Sie mehr darüber wissen möchten, dann sollten Sie sich die Anleitung von AmiTCP mal ein wenig genauer zu Gemüte führen. =;^)

Um weitere Informationen über die Optionen bzw. die durch Scout benutzbaren Befehle zu erhalten, siehe auch

Optionen  
 und

ARexx- und Shell-Befehle

.

HINWEIS: Achten Sie aus Sicherheitsgründen stets darauf, daß auch der entfernte Rechner eine aktuelle Scout-Version benutzt, um Inkompatibilitäten zu vermeiden.

## 1.36 scout-39.guide/Scout without MUI

Scout ohne MUI  
 =====

Scout bietet dem Benutzer die Möglichkeit, fast alle über die grafische Benutzungsoberfläche angebotenen Funktionen auch über die Shell zu verwenden, wobei MUI von Scout dann natürlich nicht benötigt wird.

Demzufolge müssen Sie MUI nicht unbedingt installiert haben, um Scout benutzen zu können! Wenn Sie allerdings eine grafische Benutzungsoberfläche bevorzugen, kommen Sie bei Scout nicht um MUI herum.

## 1.37 scout-39.guide/Options

## Optionen

\*\*\*\*\*

Für das Programm stehen ein paar Optionen zur Verfügung, die Sie benutzen können, wenn Sie das Programm starten. Diese Optionen können als Shell-Parameter oder als Tool Types von der Workbench benutzt werden. Dieser Abschnitt soll Ihnen den Verwendungszweck der Optionen erläutern.

Beispiel: In einer Shell werden die Optionen wie folgt benutzt:

```
l> scout option(s)
```

### ICONIFIED

Format: ICONIFIED

Wird diese Option verwendet, dann startet Scout iconifiziert.

### PORTNAME

Format: PORTNAME=portname

Der ARexx-Port von Scout kann mit Hilfe dieser Option in portname umbenannt werden. Wird diese Option nicht benutzt, dann bekommt der ARexx-Port von Scout den Namen SCOUT.X, wobei das X die Nummer der Scout-Inkarnation angibt.

### TOOLPRI

Format: TOOLPRI=value

Diese Option erlaubt es Ihnen, die Task-Priorität von Scout auf einen bestimmten Wert value zu setzen. Dieser Wert value darf nur Werte von -128 bis 127 annehmen.

### STARTUP

Format: STARTUP=command

Als Parameter kann dieser Option entweder der Name eines ARexx-Skripts oder ein ARexx-Befehl übergeben werden. Beide (das Skript oder der Befehl) werden beim Start von Scout ausgeführt.

Auf diese Weise kann zum Beispiel bei jedem Start des Programmes das Tasks-Fenster automatisch geöffnet werden. Dafür braucht nur der Befehl OpenWindow Tasks entweder der Option STARTUP übergeben werden oder in dem angegebenen ARexx-Skript enthalten sein.

### INTERVALTIME

Format: INTERVALTIME=time

Diese Option erlaubt es, die Intervallzeit einzustellen, an der die Liste der Tasks regelmäßig erneuert wird, wenn die CPU-Funktion gewählt wurde.

### CPUDISPLAY

Format: CPUDISPLAY=value

Durch die Variable value ist es möglich den Zustand des

Cycle-Gadgets, das sich im Tasks-Fenster befindet, einzustellen.  
(Siehe auch

CPU  
.)

\* 1 bedeutet CPU: full

\* 2 bedeutet CPU: in %

#### HOST

Format: HOST=hostname

Möchten Sie via AmiTCP auf einen anderen Rechner zugreifen, dann geben Sie hier bitte als hostname den Namen des gewünschten Rechners an.

#### USER

Format: USER=username

Diese Option dient dazu, um mit username den Namen des Accounts auszuwählen, über den Sie die Systemstrukturen des anderen Rechners verwalten möchten.

#### PASSWORD

Format: PASSWORD=password

Hier sollten Sie das notwendige Passwort angeben, das für das Einloggen an dem anderen Rechner notwendig ist.

#### COMMAND

Format: COMMAND=commandline

Diese Option, die als Shell-Option auch ohne das Schlüsselwort COMMAND benutzt werden kann, bietet Ihnen die Möglichkeit, einen der vielen Befehle zu benutzen, die Scout Ihnen via ARexx und Shell zur Verfügung stellt.

Siehe auch

ARexx- und Shell-Befehle  
.

#### SINGLEWINDOWS

Format: SINGLEWINDOWS

Diese Option sorgt dafür, daß jeweils nur ein Listenfenster und ein Detailfenster geöffnet sind. Leute, die nicht gewohnt sind, mit vielen Fenstern zu arbeiten, werden diese Option wohl zu schätzen wissen. =|^)

#### SORT#?TYPE

Format: SORT#?TYPE=number

Viele Listen, die von Scout angezeigt werden, können nach bestimmten Kriterien sortiert werden. Das jeweilige Sortierkriterium kann durch ein Cycle-Gadget ausgewählt werden.

SORT#?TYPE steht prinzipiell für jede der folgenden Optionen:

---

SORTLIBRARIESTYPE, SORTDEVICESTYPE,  
 SORTRESOURCESTYPE, SORTTASKSTYPE, SORTPORTSTYPE,  
 SORTCOMMANDSTYPE, SORTASSIGNSTYPE, SORTLOCKSTYPE,  
 SORTCOMMODITIESTYPE, SORTSCREENMODETYPE und SORTCLASSESTYPE.

SORT#?TYPE wird eine Dezimalzahl number übergeben, die für das jeweilige Sortierkriterium steht. Das erste (oberste) der jeweils zur Verfügung stehenden Sortierkriterien wird per Voreinstellung benutzt.

Als Beispiel folgen hier nun die jeweiligen Nummern und deren Bedeutung für die Liste der Tasks:

SORTTASKSTYPE=1: Die Taskliste wird nach Tasknamen sortiert.  
 SORTTASKSTYPE=2: Die Taskliste wird nach Prioritäten sortiert.

## 1.38 scout-39.guide/Commands

ARexx- und Shell-Befehle

\*\*\*\*\*

Bei Scout gibt es zwei Arten von Befehlen:

1. Befehle, die nur als Shell-Parameter von Scout zur Verfügung stehen
2. Befehle, die zusätzlich auch über die ARexx-Schnittstelle aufgerufen werden können

ARexx-Schnittstelle:

-----

MUI gibt jeder seiner Applikationen automatisch eine ARexx-Port (ARexx-Schnittstelle). Demnach besitzt Scout also auch einen ARexx-Port, der normalerweise den Namen SCOUT.X hat, wobei das X die Nummer der Programm-Inkarnation angibt.

Der jeweilige Name des ARexx-Ports jeder Scout-Inkarnation wird auch in dem Fenster angezeigt, welches Sie durch die Auswahl des Project/About-Menüpunktes erhalten.

Verwendung von Tasknamen:

-----

Ein Task oder ein Prozess, der von einer Shell aus gestartet wurde und sich nicht abgekoppelt hat, hat meistens einen Namen wie Background CLI oder CLI Process. Scout verwendet in der Task-Liste in einem solchen Fall nicht den 'richtigen' Namen des Tasks, sondern den Namen des jeweils ausgeführten Programmes.

Beispiel: Starten Sie zum Beispiel das Programm DH0:Debug/Sushi ohne den Befehl run, dann wird bei Scout als Taskname DH0:Debug/Sushi angezeigt.

Einige Befehle von Scout erwarten als Parameter auch einen Tasknamen. Dieser Taskname muß auf die gleiche Weise angegeben werden, wie er bei Scout angezeigt wird.

Verwendung von Adressen:  
-----

Viele der folgenden Befehle benötigen als Parameter die Adressen bestimmter Strukturen. Diese Adressen können als hexadezimale Zahlen mit im Befehlsaufruf angegeben werden.

Beispiel: Die folgenden drei Aufrufe sind syntaktisch korrekt:

1. scout FreezeTask AmiTCP:AmiTCP
2. scout FreezeTask 0x00204508
3. scout FreezeTask \$00204508

Der erste Aufruf friert den Prozess AmiTCP:AmiTCP ein, sofern dieser überhaupt vorhanden ist. Die beiden anderen Aufrufe können nur erfolgreich ausgeführt werden, wenn jeweils ein Task existiert, der an der Adresse \$00204508 im System zu finden ist.

Befehle via Shell  
=====

Help

Format: Help

Dieser Befehl Help, der keine Parameter benötigt, ist wohl der wichtigste der nun folgenden Befehle. Er veranlaßt Scout, die Liste der verfügbaren Befehle auszugeben. =:^)

Die nun folgenden 18 Befehle sind dazu da, dem Benutzer alle Listen, die Scout anbietet, auch in der Shell auszugeben. Dadurch ist es nicht mehr unbedingt erforderlich, MUI zu installieren, wenn man Scout benutzen möchte. Möchte man allerdings die vielen Fenster von Scout benutzen, kommt man um MUI nicht herum!

Für jeden dieser Befehle steht auch eine Kurzform zur Verfügung, die jeweils hinter dem Befehl in Klammern zu finden ist.

Hier also die Befehle, die jeder für sich eine Liste ausgeben:

Allocations (a), BoopsiClasses (b), Commands (c), Devices (d), Timer (e), Fonts (f), Assigns (g), InputHandlers (h), Interrupts (i), LowMemory (j), Commodities (k), Libraries (l), Memory (m), Mounts (n), Locks (o), Ports (p), Residents (r), Semaphores (s), Tasks (t), Resources (u), Vectors (v), Windows (w), Expansions (x), System (y) und ScreenMode (z).

Beispiel: Um die Liste der Ports in der Shell auszugeben, müssen Sie einfach in der Shell scout ports oder scout p eingeben.

Befehle via ARexx und Shell  
=====

Dieser Abschnitt stellt die Befehle vor, die als ARexx-Befehl und als Shell-Parameter zur Verfügung stehen.

#### FindTask

Format: FindTask task

Mit diesem Befehl kann festgestellt werden, ob ein bestimmter Task im System vorhanden ist. Er liefert als Ergebnis die Adresse des Tasks task, sofern dieser gefunden wurde. Als Variable task kann entweder der Name eines Tasks oder eine Adresse angegeben werden.

#### FreezeTask

Format: FreezeTask task

Der Task task wird von Scout eingefroren. Er ist danach zwar noch in der Task-Liste zu finden, bekommt aber keine Rechenzeit mehr vom System. Die Variable task entspricht einem Tasknamen oder der Adresse eines Tasks.

#### ActivateTask

Format: ActivateTask task

Der eingefrorenen Task task kann durch diesen Befehl wieder aktiviert werden. Für die Variable task ist ein Taskname oder eine Adresse zu wählen.

#### RemoveTask

Format: RemoveTask task

Mit diesem Befehl wird der Task mit dem Namen oder der Adresse task unwiderruflich aus dem System entfernt.

#### BreakTask

Format: BreakTask task

Dem Task task wird mit Hilfe dieses Kommandos ein Signal geschickt, das dem Drücken von CTRL-C bzw. CTRL-D entspricht. Viele Programme reagieren auf dieses Signal, indem sie sich selbständig beenden. Als Variable task kann entweder der Name eines Tasks oder eine Adresse angegeben werden.

#### SignalTask

Format: SignalTask task hexsignal

Hiermit kann dem Task task ein gewähltes Signal hexsignal (bzw. eine Signalmaske) zugeschickt werden. Dieses Signal muß als Hexadezimalzahl (mit vorangestelltem 0x oder \$) angegeben werden.

Beispiel: Das Kommando SignalTask scout 0x001000 sendet dem Scout-Prozess ein CTRL-C, worauf dieser sein Dasein beendet.

#### SetTaskPri

Format: SetTaskPri task priority

Der Task task bekommt mit Hilfe dieses Befehles die Priorität priority. Die Variable task entspricht einem Tasknamen oder der Adresse eines Tasks.

---

#### RemovePort

Format: RemovePort port

Der Port port wird von Scout aus dem System entfernt. Für port kann entweder der Name des zu entfernenden Ports oder dessen Adresse gewählt werden.

#### GetLockNumber

Format: GetLockNumber lockpattern

Dieses Kommando gibt die Anzahl der Lock-Einträge zurück, deren Pfade mit dem Namensmuster lockpattern übereinstimmen. So kann über ARexx nachgeschaut werden, ob noch auf ein bestimmtes File zugegriffen wird.

#### RemoveLocks

Format: RemoveLocks lockpattern

Alle Locks werden aus dem System entfernt, deren Pfade mit dem Namensmuster lockpattern übereinstimmen. Bei diesem Kommando ist höchste Vorsicht geboten! Will ein Programm einen Lock entfernen, der schon von Scout entfernt wurde, dann stürzt mit großer Wahrscheinlichkeit der Rechner ab.

#### RemoveLock

Format: RemoveLock lockaddress

Der Lock mit der Adresse lockaddress wird aus dem System entfernt.

#### FindNode

Format: FindNode nodetype nodename

Dieser Befehl erlaubt es Ihnen, eine Struktur nodename zu finden, die einen bestimmten Nodetypen nodetype besitzt.

Die Variable nodetype kann folgende Werte haben: LIBRARY, DEVICE, RESOURCE, MEMORY, SEMAPHORE, PORT oder INPUTHANDLER.

Beispiel: Wenn Sie die Adresse der dos.library bekommen möchten, müssen Sie den Befehl wie folgt aufrufen:

```
FindNode LIBRARY 'dos.library'
```

#### GetPriority

Format: GetPriority nodeaddress

Dieser Befehl liefert die Priorität einer Struktur, die folgenden Typ haben kann: Task, Library, Device, Resource, Port, Resident, Inputhandler, Interrupt, Semaphor oder ein Element der Memory-List.

Die Struktur müssen Sie dabei durch ihre Adresse nodeaddress auswählen, die Sie z.B. durch das ARexx-Kommando FindNode erhalten.

Beispiel: Die folgenden ARexx-Befehle beschaffen die Priorität Ihres Grafik-Speichers und legen sie in der Variablen pri ab:

```
FindNode MEMORY 'chip memory'  
addr = result  
GetPriority addr  
pri = result
```

#### SetPriority

Format: SetPriority nodetype nodename priority

Wenn Sie die Priorität einer Struktur nodename ändern möchten, können Sie dafür dieses Kommando benutzen. Wiederum kann die Variable nodetype folgende Werte haben: LIBRARY, DEVICE, RESOURCE, MEMORY, SEMAPHORE, PORT oder INPUTHANDLER. Die Variable priority muß dafür von Ihnen die Priorität bekommen, die die Struktur nodename bekommen soll.

#### CloseLibrary

Format: CloseLibrary library

Die von Ihnen mittels der Variablen library ausgewählte Library wird einmal geschlossen. Die Variable library sollte dafür mit dem Namen oder der Adresse der zu schließenden Library versehen werden.

#### RemoveLibrary

Format: RemoveLibrary library

Die durch ihren Namen oder ihre Adresse ausgewählte Library library wird geschlossen.

#### RemoveDevice

Format: RemoveDevice device

Das durch seinen Namen oder seine Adresse ausgewählte Device device wird geschlossen.

#### RemoveResource

Format: RemoveResource resource

Die durch ihren Namen oder ihre Adresse ausgewählte Resource resource wird geschlossen.

#### ObtainSemaphore

Format: ObtainSemaphore semaphore

Hierdurch wird dem System vorgegaukelt, daß das Gerät, das File oder wofür der Semaphor semaphore sonst eingerichtet wurde, von einem Programm mehr benutzt wird, als vorher. Die Variable semaphore kann dabei entweder den Namen oder die Adresse des Semaphors enthalten.

#### ReleaseSemaphore

Format: ReleaseSemaphore semaphore

Sollte ein Semaphor gerade benutzt werden, so machen Sie dem System mit dieser Funktion weis, daß ein Programm weniger das dem Semaphor entsprechende Gerät benutzt. Ein Programm, das den Semaphor beachtet, kann so eventuell versuchen, ein weiteres Mal

auf das entsprechende Gerät zuzugreifen.

#### RemoveSemaphore

Format: RemoveSemaphore semaphore

Der durch seinen Namen oder seine Adresse ausgewählte Semaphore semaphore wird mit Hilfe dieses Befehles aus dem System entfernt.

#### RemoveInpuhandler

Format: RemoveInpuhandler inputhandler

Der Inpuhandler inputhandler, den sie durch seinen Namen oder seine Adresse ausgewählt haben, wird aus dem System entfernt.

#### FindResident

Format: FindResident resident

Mit diesem Befehl kann festgestellt werden, ob eine bestimmte residente Struktur im System vorhanden ist. Er liefert als Ergebnis die Adresse der residenten Struktur resident, sofern diese gefunden wurde. Als Variable resident kann entweder der Name oder die Adresse einer residenten Struktur angegeben werden.

#### FindInterrupt

Format: FindInterrupt interruptname

Dieser Befehl dient dazu, einen bestimmten Interrupt mit dem Namen interruptname zu finden. Wird der Interrupt gefunden, so wird seine Adresse zurückgeliefert.

#### RemoveInterrupt

Format: RemoveInterrupt interruptname

Der Interrupt interruptname wird aus dem System entfernt.

#### FlushDevs

Format: FlushDevs

Sollten sich noch Devices im System bzw. im Speicher befinden, die im Augenblick von keinem Programm mehr benötigt werden, so werden sie aus dem Speicher entfernt.

#### FlushFonts

Format: FlushFonts

Unbenutzte Zeichensätze, die von Diskette bzw. Festplatte nachgeladen wurden und nicht mehr benötigt werden, werden aus dem Speicher entfernt.

#### FlushLibs

Format: FlushLibs

Sollten sich noch Libraries im System/im Speicher befinden, die im Augenblick von keinem Programm mehr benötigt werden, so werden sie aus dem Speicher entfernt.

#### FlushAll

---

Format: FlushAll

Diese Funktion beinhaltet die Funktionen FlushDevs, FlushFonts und FlushLibs. Dementsprechend werden Devices, Libraries und Zeichensätze, die zur Zeit von keinem Programm benutzt werden, aus dem Speicher entfernt.

ClearResetVectors

Format: ClearResetVectors

Bei Gebrauch dieser Funktion werden die sechs Reset-Vektoren gelöscht (siehe auch Vectors).

PopToFront

Format: PopToFront winscr

Der Screen oder das Fenster winscr werden in den Vordergrund gebracht. Die Variable winscr kann entweder den Title des Screens/Fensters oder die Adresse des Screens/Fensters enthalten.

CloseWindow

Format: CloseWindow window

Das Fenster mit dem Titel oder der Adresse window wird geschlossen.

CloseScreen

Format: CloseScreen screen

Der Screen mit dem Titel oder der Adresse screen wird geschlossen.

CloseFont

Format: CloseFont address

Der Zeichensatz mit der Adresse address wird einmal geschlossen.

RemoveFont

Format: RemoveFont address

Der Zeichensatz mit der Adresse address wird aus dem System entfernt, sofern er von keinem Programm mehr benutzt wird bzw. oft genug geschlossen wurde.

RemoveCommand

Format: RemoveCommand address

Der residente Befehl mit der Adresse address wird aus dem System entfernt.

RemoveAssign

Format: RemoveAssign name

Mit Hilfe dieses Befehles wird der Assign mit dem Namen name aus dem System entfernt.

RemoveAssignList

---

Format: RemoveAssignList name address

Dieser Befehl sorgt dafür, daß das Verzeichnis mit der Adresse address von dem Assign mit dem Namen name entfernt wird.

#### PrintList

Format: PrintList listkey filename

Um auch via ARexx an eine der gegebenen Struktur-Listen zu kommen, wurde Scout der Befehl PrintList beigelegt. Die Liste, deren Tastaturkürzel der Variablen listkey übergeben wird, wird in das File filename ausgegeben.

#### OpenWindow

Format: OpenWindow windowid

Mit diesem Kommando sind Sie in der Lage, alle Fenster über ARexx zu öffnen, die über das Hauptfenster von Scout durch das Betätigen eines Gadgets geöffnet werden können.

Die Fensteridentifikation windowid besteht aus dem gleichen Text, der auch auf den Gadgets im Hauptfenster zu finden ist.

Beispiel: Wird das Kommando OpenWindow 'Resident Cmds' zu Scouts ARexx-Port geschickt, dann wird das Fenster mit der Liste der residenten Befehle geöffnet.

Sollte das Fenster schon geöffnet worden sein, dann wird es nach vorn geholt, und die jeweilige Liste wird neu eingelesen.

Aus der diesem Befehl zugedachten Aufgabe wird ersichtlich, daß dieser Befehl keinerlei Wirkung hat, sollte er als Shell-Parameter aufgerufen worden sein. Die grafische Oberfläche von Scout steht dort eben nicht zur Verfügung.

#### CxAppear

#### CxDisappear

#### CxEnable

#### CxDisable

#### CxKill

#### CxListChg

#### CxUnique

Format: Cx... name

Hiermit wird dem Commodity mit dem Namen name das entsprechende Kommando zugesendet. Ob und was dann geschieht, hängt von dem Commodity selbst ab.

#### RemoveCx

Format: RemoveCx commodity

Das entsprechende Commodity wird aus der Liste entfernt. Dies ist jedoch nur als Notfalllösung gedacht, wenn CxKill nicht funktionierte.

#### SetCxPri

Format: SetCxPri commodity priorität

Setzt die Priorität des Commodities.

RemoveClass

Format: RemoveClass class

Die entsprechende BOOPSI-Klasse wird geschlossen, wenn kein Objekt und keine Unterklasse existiert.

## 1.39 scout-39.guide/Updates

Wie und wo bekommt man Updates?

=====

Die neueste Version von Scout sollte immer auf dem AmiNet im Verzeichnis util/moni zu finden sein. Wahrscheinlich wird sie sich dadurch automatisch etwas später auch in aktuelleren Public Domain Sammlungen befinden.

Auch finden Sie die aktuelle Version stets auf meiner Homepage:  
<http://www.is-koeln.de/einwohner/shred/>

## 1.40 scout-39.guide/Credits

Wem ich zu danken habe

=====

Nun haben wir noch ein paar Leuten zu danken, die uns bei der Entwicklung von Scout auf die unterschiedlichsten Weisen behilflich waren, als da wären:

- \* Klaus 'gizmo' Weber, der dieses Programm ein wenig unter die Lupe genommen hat und für Attes Probleme bei der Entwicklung von Scout (es waren nicht wenige) meist ein freies Ohr hatte,
- \* Christian 'cosinus' Stelter, der es erlaubt hat, seine ganzen Manuals zu benutzen,
- \* Stefan Stuntz für sein MagicUserInterface, das uns viel Arbeit erspart hat,
- \* allen, die die Bezeichnung Giftware ernst genommen hatten,
- \* den ganzen Leuten, die Scout getestet und uns Bugs oder neue einzubauende Features gemeldet haben und es (hoffentlich auch weiterhin) noch tun: Kai 'wusel' Siering, Martin Hauner, Peter Meyer, Karl 'Charly' Skibinski, Michael 'Mick' Hohmann, Thore Böckelmann, Bernardo Innocenti, Daniel Lundberg, ...

und zum guten Schluß

\* all den anderen, die wir evtl. vergessen haben, die uns Bugs, Anregungen und konstruktive Kritik zu Gehör gebracht haben.

## 1.41 scout-39.guide/Author Info

Wie erreicht man die Autoren?

=====

Wenn Sie Fragen, Verbesserungsvorschläge, Bug Reports oder Dinge dieser Art haben, dann können Sie mich unter der folgenden Adresse erreichen:

Richard Körber  
Hornstraße 20  
51465 Bergisch Gladbach  
- Germany -

E-Mail: shred@chessy.aworld.de  
richard.koerber@koeln.netsurf.de

Schreiben Sie E-Mails, wenn irgend möglich.

Andreas Gelhausen können Sie erreichen unter:

Andreas Gelhausen  
Graf Spee Str. 23b  
26123 Oldenburg  
- Germany -

E-Mail: atte@crash.north.de

Bitte senden Sie ihm keine Bug-Reports, Vorschläge und ähnliches. Für Leute, die unbedingt noch ein Geschenk loswerden möchten, ist er jedoch die richtige Adresse.

Das war's! =:^)

## 1.42 scout-39.guide/Index

Stichwortverzeichnis

\*\*\*\*\*

Adresse des Autors  
Author Info

---

Adressen, Verwendung von Commands	
Allocations	Allocations
AmiTCP	AmiTCP
Analyse des Systems System	
ARexx-Befehle	Commands
ARexx-Schnittstelle	Commands
Assigns	Assigns
Autor	Author Info
Bildschirm-Modis	ScreenMode
Boards	Expansions
BoopsiClasses	BoopsiClasses
CLI Optionen	Options
Commodities	Commodities
Copyright	Copyright
Copyright	Copyright
Danksagungen	Credits
Devices	Devices
DISKFONT	Fonts
Einleitung	Introduction

---

---

Ereignisse	InputHandlers
Erweiterungskarten	Expansions
Expansions	Expansions
Fenster	Windows
Festplatten	Mounted Devs
Fonts	Fonts
FreeWare	Copyright
Generelle Benutzung	Using Scout
Haftungseinschränkung	Copyright
Handler, LowMemory	LowMemory
Hardware	Expansions
Hauptfenster	Using Scout
Hersteller	Expansions
Identify	Identify
Inhalt des Pakets	Copyright
Input Events	InputHandlers
Inputhandler	InputHandlers
Installation	Installation
Interrupts	Interrupts

---

---

Keine Garantie	Copyright
Kopieren	Copyright
Laufwerke	Mounted Devs
Libraries	Libraries
Locks	Locks
Logische Verzeichnisse	Assigns
LowMemory	LowMemory
MagicUserInterface	MUI
Manufacturer	Expansions
Markenzeichen	Copyright
Memory	Memory
Mounted Devices	Mounted Devs
MUI	MUI
Nutzungseinschränkung	Copyright
Optionen	Options
Ports	Ports
Processes	Tasks
Programmversion	Updates
Prozesse	Tasks

---

---

RAM Pointer Count	Devices
Rechtliche Dinge	Copyright
Resident Commands	Resident Cmds
Residente Befehle	Resident Cmds
Residente Strukturen	Residents
Residents	Residents
Ressourcen-Belegungen	Allocations
Resources	Resources
Ressourcen	Resources
ROMFONT	Fonts
RPC	Devices
ScreenMode	ScreenMode
Screens	Windows
Semaphore	Semaphores
Semaphores	Semaphores
Shell Optionen	Options
Shell-Befehle	Commands
Speichersegmente	Memory
System	System

---

System-Erweiterungen	
Expansions	
Systemanforderungen	
System Requirements	
Tasknamen, Verwendung von	
Commands	
Tasks	Tasks
TCP/IP	AmiTCP
Timer	Timer
Tool Types	Options
Updates	Updates
VBR	Vectors
Vectors	Vectors
Vektoren	Vectors
Verteilung	Copyright
Vertical blank interrupt	
Interrupts	
Was ist Scout?	Introduction
Windows	Windows
Zeichensätze	Fonts

---