
Update EVOLUTION 3.0

**Update-Satz für den
SCSI-II-Controller
für AMIGA 2000/1000/500**

Hochkompatibles Treibersystem für den
MacroSystem-HighSpeed-SCSI-II-Controller

Installation und Bedienung

Ein MacroSystem-Product

Inhaltsverzeichnis

1	Allgemeines	4
2	Hardwareseitige Installation des Updates	5
2.1	Installation eines Update-Paketes im Multi-Evolution	5
2.2	Installation eines Update-Paketes im Evolution 2000	5
3	Softwareseitige Installation des Updates	6
3.1	Neuformatierung - Ja oder Nein?	6
3.2	Das Installationsprogramm unter Kickstart 1.3	6
3.3	Das Installationsprogramm unter Kickstart 2.0	7
3.4	Die automatische Installation	7
3.5	Die manuelle Installation	7
3.5.1	Das Fenster 'Angaben zum Gerät'	7
3.5.2	Das Fenster 'Partitionieren...'	8
3.5.3	Das Fenster 'Weitere Einstellungen'	8
3.6	Die Menüleiste	9
3.6.1	Das Menü 'Projekt'	9
3.6.2	Das Menü 'Einstellungen'	9
3.6.3	Das Menü 'Benutzer'	10
4	Die Benutzung der 'HDToolBox'	10
5	Das AREXX-Interface	10
5.1	Kurzbeschreibung der Parameterarten	11
5.2	Rückgabewerte und Fehlercodes	11
5.3	Die einzelnen ARexx-Befehle	12
6	Besonderheiten beim Betrieb von Nicht-Festplatten-SCSI-Geräten	24
6.1	Tape Streamer	24
6.2	CD-ROM-Laufwerke	24

7.4 ^{19b 1:} QuantumControl

Die beliebte Quantum-Festplattenserie läßt sich softwareseitig frei justieren. Dies betrifft den Cache-Speicher, die Unit-Nummer etc.

Dieses Programm zeigt die Serienwerte an und ermöglicht Änderungen.

Es wird gestartet per

QuantumControl [Unit-Nummer]

und fragt dann nach evtl. Änderungswünschen.

Sie sollten sich die serienmäßigen Zustände notieren, denn erfahrungsgemäß sind die Werkseinstellungen sehr sinnvoll.

Ansonsten ist Optimieren durch freies Herumspielen möglich!

7.5 BindSCSI

Dieses Programm erlaubt die Einbindung von SCSI-Geräten, die sich nicht bei Systemstart als SCSI-Gerät zu erkennen gegeben haben und somit normalerweise keine legale Ansprechmöglichkeit hätten. Auch Geräte, die der Anwender aufgrund Geräusentwicklung, Verzögerung beim Systemstart etc. nicht immer eingebunden haben möchte, können per 'bindscsi' nachträglich im Bedarfsfall eingebunden werden.

Bei Start des Evolution-Installationsprogrammes wird 'bindscsi' übrigens nicht benötigt.

Etwa Streamer etc. seien hier genannt.

BindSCSI [Unit-Nummer]

also etwa BindSCSI 3 für die Unit-Nummer 3 ist hier die Syntax und führt zur Einbindung des Gerätes an dieser Adresse.

7.6 GetD

Dieses Programm ermittelt die Größe und Beschaffenheit des SCSI-Gerätes an der angegebenen Unit-Nummer, also etwa 'GetD 3' für Unit Nummer 3.

Im wesentlichen wird ein solcher Befehl auch von 'Read characteristics from drive' unter HDToolBox und von der Installationssoftware durchgeführt, jedoch rechnet GetD auch die optimal raumausnutzende Kombination von Zylindern, Sektoren pro Track und Köpfen aus.

Dieses Zusatzprogramm ist vor allem in Verbindung mit SCSI-Software von Drittherstellern sinnvoll. Im Evolution-Installationsprogramm ist es bereits implementiert und muß nicht extra aufgerufen werden.

Sollten noch offene Fragen oder Probleme existieren, so schreiben Sie uns oder rufen Sie uns an. Wir werden dann unser Möglichstes tun, um Ihnen zu helfen. Bei Interesse an unseren weiteren AMIGA-Produkten: Prospektmaterial kommt prompt gegen frankierten DIN A 4 - Rückumschlag!

Evolution verfügt über einen internen Burst-Modus, der die Geschwindigkeit der Festplatten unter 68000 stark erhöhen kann. Unter 68020/30/40 muß der Burst-Mode unter Umständen wieder deaktiviert werden. Mit BURST wird aktiviert, mit NOBURST deaktiviert.

Das Format des Befehls lautet also

HDParms [Unit-Nummer] CACHE/NOCACHE BURST/NOBURST FORMAT/NOFORMAT CHANGE [Sekunden]

also etwa

HDPARMS 0 CACHE NOFORMAT NOBURST CHANGE 5

für eine Festplatte in Unit 0, die mit Cache, aber ohne Burst bedient werden soll, auf die ein Auto-Laufwerkswechsel aktiviert werden soll, der alle 5 Sekunden den Status des Laufwerks kontrolliert.

7.2 LowLevel

Dieser Befehl hat die Syntax

LowLevel [Unit-Nummer]

also etwa 'LowLevel 6' zum Formatieren der SCSI-Festplatte Nummer sechs.

Dieser Format führt ein sattelfestes Pre-Format der Festplatte durch, wobei selbstverständlich alle Daten verlorengehen. Dies ist bei einigen SCSI-Geräten (etwa SCSI-Diskettenlaufwerke, Wechselplatten und an Fremdcontrollern vorformatierten Festplatten) empfehlenswert. Nicht alle Laufwerke reagieren auf dieses Kommando und können nicht vorformatiert werden. Dies ist zum Beispiel bei vielen Quantum-Typen der Fall.

7.3 Park und Start

Diese Befehle mit der Syntax

Park [Unit-Nummer]

Start [Unit-Nummer]

also etwa 'Start 3' zum Starten der an Unit 3 installierten Festplatte, dienen dem Stoppen und Re-Starten von Festplatten im laufenden Betrieb.

Nicht alle Festplatten verstehen diese speziellen SCSI-Befehle, aber bei Quantums und Syquests sollte es keine Probleme geben.

Ist das betroffene Laufwerk als wechselbar angegeben, so führt ein 'Park' zum Entfernen des Icons von der Workbench und Replazieren als 'Busy'.

Ein 'Start' fährt den Motor wieder auf Touren, die Festplatte ist wieder einsatzbereit.

ACHTUNG: Wenn Sie Ihre Hauptfestplatte parken, so benötigen Sie das Programm 'Start' auf einem anderen Medium (Diskette o.ä.), um diese Festplatte wieder ohne Reset betreiben zu können.

6.3 Magneto-Optische Laufwerke	24
7 Die Utilities des Evolution 3.0-Paketes	25
7.1 HDParms	25
7.2 LowLevel	26
7.3 Park und Start	26
7.4 QuantumControl	27
7.5 BindSCSI	27
7.6 GetD	27

1 Allgemeines

Als MacroSystem 1990 den Evolution-Controller für den Amiga 2000 vorstellte, war dies der Beginn der Ära 'High-Speed SCSI für Amiga'.

Seitdem hat sich an Evolution viel verändert. So manche Verbesserung und Anpassung wurde durchgeführt. Controller auch für A 500 folgten, wahlweise mit RAM-Option und interner Festplatte. Die Tests in allen bekannten Magazinen ergaben stets nur beste Beurteilungen.

Unverändert jedoch blieb die hohe Datenrate, der Grundstock für einen erfolgreichen SCSI-Controller überhaupt.

Mit der nun vorliegenden Version 3.0 wurden viele bahnbrechende und langerwartete Neuerungen realisiert. Für den Anwender nicht direkt offensichtlich ist die Tatsache, daß das komplette Treibersystem vollständig neu aufgebaut wurde, um einigen prinzipiellen Erfordernissen gerecht zu werden. Offensichtlich sind dagegen die folgenden Features:

- Vollständiger Support des Commodore-definierten "Rigid Disk" Boot-Blocks, wodurch die Festplatten/Medien frei mit vielen anderen SCSI-Controllern getauscht werden können
- Auto-Drive-Change (automatische Wechselregistrierung) für Geräte wie Syquest-Wechselplatten oder magneto-optische Wechselgeräte
- Installationssoftware für OS 2.x, streng nach den Richtlinien des User-Interface Style Guides" (Commodore-Vorgabe für die Gestaltung von Benutzeroberflächen), mit übersichtlicher Maussteuerung und AREXX-Port zur Programmierung von anderen, auch eigenen Programmen heraus
- Unterstützung für Tape Streamer, optische Wechselplatten (wiederbeschreibbare CDs) und CD-ROM-Laufwerken
- Spezielle Software zur Konfigurierung der Quantum-Festplatten, frei setzbare Caches und sogar permanente softwareseitige Unit-Nummern-Verstellung
- Die Abschaltung des Systems wurde von der Maustasten-Bedienung auf die 'Del'- und die 'Help'-Taste umgelegt. Da das AmigaOS 2.0 ebenfalls die Maustasten abfragt, ergaben sich Kollisionen. Die 'Del'-Taste unterbricht die Suche nach Festplatten, die 'Help'-Taste verhindert die Suche ganz. Wenn beide Tasten gleichzeitig vom Rechnerstart an gedrückt werden, wird Evolution ganz abgeschaltet.

Weiterhin wurde das gesamte System in Punkto Zuverlässigkeit und Kompatibilität verbessert.

Das Update gibt es in zwei Versionen:

1. Für den Evolution A 2000 und 1000/500 ohne RAM-Option
2. Für den MultiEvolution A 500 (mit RAM-Option)

Achtung, die beiden Update-Sätze sind einander zwar ähnlich, sind aber nicht austauschbar. Sollten Sie nicht die passende Version erhalten haben (lesen Sie den Text auf Ihrer Rechnung), so setzen Sie sich sofort mit Ihrem Händler in Verbindung!

Diese Anleitung soll die Installation des Updates sowohl hard- als auch softwareseitig erläutern. Bitte lesen Sie diese Anleitung komplett, um Fehlbedienungen zu vermeiden, die Ihnen eventuell einen nicht wiederherzustellenden Datenverlust "bescheren" könnten!

notwendigen SCSI-Kommandos nicht beherrschen. Diese Geräte bedürfen der Festlegung 'per Hand', das heißt, manuelles Eintragen ist notwendig.

Besonders interessant sind hier die Ricoh-Laufwerke, die sehr schnell sind und die auch als 3.5"-Gerät existieren. Diese Geräte sind mit Evolution 3.0 intensiv getestet worden und sind aus den Tests der Fachzeitschriften als Testsieger hervorgegangen.

Auch als Datensicherungsgerät sind sie sehr zu empfehlen.

7 Die Utilities des Evolution 3.0-Paketes

Zum Betrieb des Treibers sind keine weiteren Utilities notwendig, dennoch liegen dem Paket einige solche Programme bei.

Diese sollen im folgenden kurz erläutert werden.

7.1 HDParms

Bei diesem Befehl handelt es sich wohl um das wichtigste Utility im Paket. Vielfach kann die volle Performance erst durch die Verwendung von HDParms erreicht werden. Viele Anwender werden den Befehl mit der entsprechenden Einstellung in der 'startup-sequence' bzw. 'user-startup' plazieren wollen.

Wie bereits erwähnt, benutzt Evolution 3.0 in der aktuellen Version den RDB-Bootblock. Dieser erlaubt aufgrund seiner festen Definition keine Zusatzparametereinstellung im Bootblock.

Deshalb wurde HDParms zur Einstellung der Evolution-spezifischen Parameter entwickelt.

HDParms erlaubt die folgenden Einstellungen:

- **Formatierschutz**

Häufig kommt es vor, daß der Anwender beim Versuch, eine Diskette zu formatieren, statt df0: dh0: tippt. Dagegen kann der Anwender sich schützen, indem er die Festplatte mit dem switch 'NOFORMAT' versieht.

- **Drive Change**

EVOLUTION 3.0 kann den Laufwerkswechsel von Festplatten automatisch erkennen. Dies arbeitet prinzipiell nur mit gleich formatierten Medien/Festplatten. Beim Systemstart wird festgestellt, ob ein Laufwerk als Wechsellaufwerk arbeitet (Syquest etc.). In diesem Fall wird das Laufwerk automatisch mit einem Drive Change versehen. Für jedes Wechsellaufwerk wird ein Task gestartet, der in festen Zeitabständen den Status des Laufwerks überprüft und bei einer Veränderung Aktionen einleitet.

Per Option 'CHANGE' können auch normale Laufwerke mit automatischem Wechsel versehen werden. Nach dem CHANGE kann auch die Zeit zwischen den Tests festgesetzt werden. Dies ist ein Wert zwischen 0 und 99, 0 = ständiger Check, 99 = Alle 99 Sekunden erfolgt eine Überprüfung. NOCHANGE schaltet die automatische Laufwerkswechselung ab.

- **Cache**

Das Evolution 3.0-System verfügt über ein Cache-System, welches die Geschwindigkeit der Laufwerke erhöht. Das Caching kann per 'CACHE' aktiviert und per 'NOCACHE' wieder deaktiviert werden.

- **Burst**

6 Besonderheiten beim Betrieb von Nicht-Festplatten-SCSI-Geräten

Bisher war Evolution auf den Betrieb von Festplatten beschränkt. Evolution 3.0 kann darüber hinaus auch Tape Streamer, CD-ROM-Laufwerke, Magneto-Optische Datenträger etc. bedienen. Zum Teil wird allerdings eine Software benötigt, die diese Geräte ansteuert.

Auf die Besonderheiten soll hier näher eingegangen werden.

6.1 Tape Streamer

Tape Streamer sind Bandlaufwerke, die zur Datensicherung verwendet werden. Sie können prinzipiell nicht als AmigaDOS-Laufwerk verwendet werden. Dafür sind sie eine preiswerte und schnelle Möglichkeit zur risikofreien Datensicherung.

Tape Streamer bedürfen einer Steuersoftware. Diese ist nicht Bestandteil des Evolution-Updates, da eine solche Steuersoftware für sich allein ein großes Projekt ist. Gängige Backup-Programmpakete wie Quarterback 5.x oder AmiBack können Tape Streamer ansteuern.

Da viele Tape Streamer sehr lange Startzeiten haben und nur gelegentlich verwendet werden, werden viele Anwender sie nur bei Bedarf über den Befehl "bindscsi" einbinden.

Dazu installieren Sie die Festplatten ohne angeschlossenen Tape Streamer. Wenn diese Installation beendet ist, schalten Sie den Rechner aus, schließen den Streamer an (Unit-Nummern beachten) und starten erneut. Wenn Sie die Einbindung wünschen, starten Sie "bindscsi [Unit-Nummer]" per CLI-Aufruf. Danach kann Quarterback oder AmiBack den Streamer problemlos ansprechen.

6.2 CD-ROM-Laufwerke

CD-ROM-Laufwerke sind CD-Spieler, die Daten in Form von speziellen CDs (im ISO 9960-Format oder im HighSierra-Format) einlesen können. Sie sind Nur-Lese-Geräte.

Zum Betrieb benötigt man ein sogenanntes FileSystem. Mit Evolution getestet wurde das FileSystem von Ralph Babel, welches Sie direkt bei MacroSystem oder im Fachhandel beziehen können.

Für CD-ROM-Laufwerke gilt ebenfalls, daß die sehr langen Startzeiten störend sein können und der Anwender die Einbindung bei Bedarf mit "bindscsi" vornehmen kann.

Dazu installieren Sie die Festplatten ohne angeschlossenes CD-ROM-Laufwerk. Wenn diese Installation beendet ist, schalten Sie den Rechner aus, schließen das CD-ROM-Laufwerk an (Unit-Nummern beachten) und starten erneut. Wenn Sie die Einbindung wünschen, starten Sie "bindscsi [Unit-Nummer]" per CLI-Aufruf. Danach kann das Laufwerk per Mount-Befehl (siehe Dokumentation zum FileSystem) eingebunden und gelesen werden.

6.3 Magneto-Optische Laufwerke

Hier handelt es sich um wiederbeschreibbare CD-Systeme. Die CDs sind zumeist nicht mit den Medien eines CD-ROM-Gerätes kompatibel. Es handelt sich aber um relativ schnelle und preiswerte Medien, die zudem eine relativ hohe Datensicherheit gewährleisten. Sie können, anders als Tape Streamer, als vollwertiges AmigaDOS-Laufwerk benutzt werden.

Die Installation dieser Geräte bedarf keiner zusätzlichen Software. Einige Geräte können jedoch nicht von der Installationssoftware hinsichtlich Größe und Parametern bestimmt werden, da sie die

2 Hardwareseitige Installation des Updates

Wichtig: Vor dem Beginn der Installation sollten Sie sicherheitshalber ein Backup (Datensicherung) durchführen. Unter Umständen ist eine Neuformatierung der Festplatte notwendig (siehe das Kapitel "Neuformatierung - Ja oder Nein?").

Die Hardwareinstallation beschränkt sich im großen und ganzen auf das Einsetzen der neuen EPROMs. Dieser Vorgang ist im folgenden, getrennt nach Controllertyp, erläutert.

2.1 Installation eines Update-Paketes im Multi-Evolution

Das Update-Paket auf V 3.0 besteht aus 2 EPROMs, 1 Diskette und dieser Anleitung. Bitte überprüfen Sie zunächst, ob alle Bestandteile vorhanden sind!

Zum Einsetzen der EPROMs müssen Sie MultiEvolution vom Amiga abziehen und das Gehäuse öffnen. Ist eine Festplatte eingebaut, so muß diese entfernt werden.

Die EPROMs befinden sich im hinteren Drittel der Platine und sind leicht zu erkennen, da sie optisch wie die beiden "neuen" des Update-Paketes aussehen.

Die EPROMs sind gesockelt und können mit einem Schraubendreher vorsichtig ausgehebelt werden. Vorsicht, verwechseln Sie nicht die alten EPROMs mit den neuen!

Die EPROMs sind als 'L' (Low) und 'H' (High) bezeichnet. "L" wird dabei links, "H" rechts eingesetzt.

Achten Sie dabei auf die Einkerbung an einer EPROM-Seite. Diese gibt die Einbaurichtung der EPROMs an. Sie muß der Einkerbung in den Sockeln entsprechen!

Bitte vergewissern Sie sich, das alle Pins der EPROMs fest und sicher im Sockel stecken. Andernfalls kann MultiEvolution natürlich nicht funktionieren.

Danach montieren Sie das System und stecken es wieder an den AMIGA an.

Nun beginnt der softwareseitige Teil der Installation.

2.2 Installation eines Update-Paketes im Evolution 2000

Das Update-Paket auf V 3.0 besteht aus 2 EPROMs, 1 Diskette und dieser Anleitung. Bitte überprüfen Sie zunächst, ob alle Bestandteile vorhanden sind!

Zum Einsetzen der EPROMs müssen Sie Evolution aus dem A 2000 ausbauen.

Die EPROMs befinden sich im vorderen Teil der Platine und sind leicht zu erkennen, da sie optisch wie die beiden "neuen" des Update-Paketes aussehen.

Die EPROMs sind gesockelt und können mit einem Schraubendreher vorsichtig ausgehebelt werden. Vorsicht, verwechseln Sie nicht die alten EPROMs mit den neuen!

Die EPROMs sind als 'L' (Low) und 'H' (High) bezeichnet. "L" wird dabei oben, "H" unten eingesetzt.

Achten Sie dabei auf die Einkerbung an einer EPROM-Seite. Diese gibt die Einbaurichtung der EPROMs an. Sie muß der Einkerbung in den Sockeln entsprechen!

Bitte vergewissern Sie sich, das alle Pins der EPROMs fest und sicher im Sockel stecken. Andernfalls kann MultiEvolution natürlich nicht funktionieren.

Danach montieren Sie das System und stecken es wieder in den Slot des Amigas.

Nun beginnt der softwareseitige Teil der Installation.

3 Softwareseitige Installation des Updates

Gegenüber früheren Updates ist bei Umstieg auf V 3.0 eine neue Installation der Software zwingend erforderlich. Grund hierfür ist die Umstellung des Bootblocks. V 3.0 benutzt den Commodore-Rigid-Disk-Block (RDB), der mit dem Evolution-Bootblock der V 2.x nicht kompatibel ist.

Nach dem Einschalten des Rechners wird das MacroSystem-Label auf dem Bildschirm erscheinen. Die Festplatte(n) werden nicht automatisch booten und der Suchvorgang erst nach 7 Durchläufen abgebrochen werden. Das liegt an der RDB-Suchroutine, die ja noch keinen RDB finden kann und deshalb nicht weiß, welche Festplatten mit welcher Partitionierung existieren.

Der Suchvorgang kann übrigens jederzeit durch gedrückthalten der 'Del'-Taste unterbrochen werden. Die Abfrage erfolgt nach dem Suchen der sechsten Unit, wenn in diesem Moment die 'Del'-Taste gedrückt ist, wird die Suche abgebrochen.

3.1 Neuformatierung - Ja oder Nein?

Da der Rigid-Disk-Block mehr Platz in Anspruch nimmt, kann es sogar erforderlich sein, die Festplatte neu zu formatieren. Dies muß dann erfolgen, wenn die erste (und evtl. einzige) Partition der Festplatte keinen Platz für den RDB mehr aufweist.

Abhängig ist dies vom LowCylinder (Start-Zylinder) der ersten Partition und vom benutzten Betriebssystem. Die folgende Tabelle soll bei der Entscheidung helfen, gilt aber nur im Regelfall, da Konstellationen existieren können, die ein Neuformat trotzdem notwendig machen.

	Start-Zylinder	
Kickstart 1.3	0 oder 1	Formatierung notwendig
	2 oder höher	Formatierung nicht notwendig
Kickstart 2.0	0	Formatierung notwendig
	1 oder höher	Formatierung nicht notwendig

Der Grund für die einfachere Umrüstung unter OS 2.x ist die Tatsache, daß unter 1.3 das FastFileSystem nicht im Kickstart vorliegt, sondern im RDB gespeichert wird. Dadurch geht ein weiterer Zylinder verloren.

Natürlich läßt sich eine Partition mit Start-Zylinder 1 nicht mehr unter 1.3 verwenden. Benutzer von Umschaltplatinen sollten deshalb eine Neuformatierung in diesem Fall in Erwägung ziehen.

Die meisten Anwender werden den Start-Zylinder auf 1 gesetzt haben, da dies der Vorgabewert in der Installationssoftware war.

Grundsätzlich gilt, daß der alte Boot-Block von der neuen Installationssoftware gelesen werden kann und so einfach in einen RDB umgewandelt werden kann. Die Feststellung des Start-Zylinders können Sie also der Installationssoftware überlassen.

3.2 Das Installationsprogramm unter Kickstart 1.3

Bei Drucklegung dieser Anleitung existierte das geplante Installationsprogramm unter Kickstart 1.3 noch nicht. Unserer Ansicht nach hat 1.3 aufgrund der grandiosen Vorteile des AmigaOS 2.0 wenig Existenzberechtigung. Dennoch wird es ein mit dem alten Betriebssystem kompatibles Programm

RC: ERROR, Gerät nicht gefunden FARAL, Number nicht im Bereich von 0 bis 6

Kommando: UNZOOMWINDOW

Schablone: START/S,STATUS/S,COMMAND/S,SELECTUNIT/S,PARAMETER/S, PARTITION/S,ADVANCEDOPTIONS/S

Zweck: Die angebebenen Fenster werden auf ihre volle Größe gebracht.

RC: ERROR, wenn angegebenes Fenster nicht geöffnet ist FATAL, wenn kein Fenster angegeben

Kommando: VERSION

Schablone: VERSION/N,REVISION/N,FULL/S,INTERNAL/S

Zweck: Test der Versions- und Revisions-Nummer oder Abrufen des Versionsstrings. Bei Angabe einer Versionsnummer wird kein Versionsstring zurückgegeben.

Beispiele: VERSION Versionsstring holen ohne Datum

VERSION FULL Versionsstring holen mit Datum

VERSION 1 Test ob die Versionsnummer größer oder gleich 1 ist.

VERSION 2 10 Test ob die Versionsnummer größer oder gleich 2 ist. Ist die Versionsnummer größer als 2, erhält man in RC den Wert 0, ist sie kleiner, den Wert 5 (Warnung). Ist sie gleich, wird getestet, ob die Revisionsnummer größer oder gleich 10 ist. Wenn nicht, erhält man ebenfalls den Wert 5 in RC

RC: WARN, wenn Versions- bzw. Revisionsnummer zu niedrig RESULT: Versionsstring wenn RC=OK

Kommando: WINDOWTOBACK

Schablone: START/S,STATUS/S,COMMAND/S,SELECTUNIT/S,PARAMETER/S, PARTITION/S,ADVANCEDOPTIONS/S

Zweck: Ein anzugebendes Fenster nach hinten bringen.

RC: ERROR, wenn Fenster geschlossen. FATAL, wenn kein Fenster angegeben wurde

Kommando: WINDOWTOFRONT

Schablone: START/S,STATUS/S,COMMAND/S,SELECTUNIT/S,PARAMETER/S, PARTITION/S,ADVANCEDOPTIONS/S

Zweck: Ein anzugebendes Fenster nach vorne holen.

RC: ERROR, wenn Fenster geschlossen FATAL, wenn kein Fenster angegeben wurde

Kommando: ZOOMWINDOW

Schablone: START/S,STATUS/S,COMMAND/S,SELECTUNIT/S,PARAMETER/S, PARTITION/S,ADVANCEDOPTIONS/S

Zweck: Die angebebenen Fenster werden auf Minimalgröße gebracht (Nur Titelleiste).

RC: ERROR, angegebenes Fenster ist nicht geöffnet FATAL, wenn kein Fenster angegeben

SETHSC Beim Aufruf ohne Parameter wird erst versucht einen RDB zu laden, im Fehlerfall wird versucht einen EVO2-Block zu laden und erst wenn selbst dies nicht möglich ist wird wie bei SETHSC DEFAULT vorgegangen.

Kommando: SETPATH

Schablone: PRESETS/K,MACROS/K,FILESYSTEM/K,USER/K

Zweck: Ein Pfad für einen der File-Requester wird geändert.

Kommando: SETPATTERN

Schablone: PRESETS/K,MACROS/K,FILESYSTEM/K,USER/K

Zweck: Der Pattern-String für einen der File-Requester wird geändert.

Kommando: SIZE *)

Schablone: MByte/A/N,FORCE/S

Zweck: Es werden Start- und Endzylinder berechnet, die eine Partition ergeben, die die gewünschte Größe hat.

Beispiel: SIZE 10 FORCE Die aktuelle Partition erhält wenn möglich eine Größe von 10 MByte. Der Parameter FORCE bewirkt, dass keine Sicherheitsabfrage erfolgt.

RC: ERROR, Auf dem aktuellen Gerät ist nicht mehr genügend Platz für die gewünschte Größe

Kommando: STARTCYLINDER *)

Schablone: Cylinder/A/N,FORCE/S

Zweck: Setzen des Startzylinders der aktuellen Partition. Ist diese Partition bereits in der Partitionsliste eingetragen, erfolgt eine Sicherheitsabfrage. Nach einer einmaligen Bestätigung dieser Sicherheitsabfrage erfolgen bei weiteren Änderungen keine weiteren.

Kommando: TESTUNIT

Schablone: Number/A/N

Zweck: Überprüfung, ob das gewünschte Gerät vorhanden ist. Dieses Kommando dient lediglich dazu, über ARexx beim UNIT Kommando eine Fehlermeldung auszuschließen.

RC: WARN, Gerät nicht gefunden

Kommando: TEXT

Schablone: NOLINE/S,STRING/F

Zweck: Ausgabe eines Strings im Hilfstext-Fenster.

Beispiele: TEXT Dies ist ein Test Der Text: "Dies ist ein Test" wird im Hilfstext-Fenster ausgegeben.

TEXT NOLINE Hallo Der Text: "Hallo" wird ohne abschließenden Zeilenvorschub im Hilfstext-Fenster ausgegeben.

Kommando: UNIT *)

Schablone: Number/A/N

Zweck: Wahl eines zu installierenden Gerätes.

geben, um den Anwendern zu helfen, die aus verschiedenen Gründen nicht auf 2.0 umsteigen können oder wollen.

Bis das oben genannte Programm erhältlich ist, kann selbstverständlich auch mit der auch unter Kickstart 1.3 arbeitenden 'HDToolBox' gearbeitet werden. Dazu lesen Sie bitte das entsprechende Kapitel in dieser Anleitung.

Kickstart 2.0 erhalten Sie im Fachhandel oder direkt bei MacroSystem. Wir empfehlen den Umstieg in jedem Fall. Der Amiga kann erst mit 2.0 zu seiner vollen Leistungsfähigkeit finden!

3.3 Das Installationsprogramm unter Kickstart 2.0

Unter Kickstart 2.0 können Sie das eigentliche und mit den vollen Features versehene Installationsprogramm nutzen. Es befindet sich im Ordner "Evolution3.0" und heißt "Evo3Install".

Nach dem Umbau der EPROMs sollten Sie von der Installationsdiskette booten. Das Programm startet automatisch. Zunächst werden Sie nun gefragt, ob Sie automatische oder manuelle Installation wünschen.

3.4 Die automatische Installation

Die automatische Installation führt die gesamten notwendigen Vorgänge selbstständig aus. Dabei geht Sie von folgenden Präferenzen aus:

- Ist ein RDB vorhanden, wird dieser komplett übernommen.
- Ist ein Evolution-Bootblock vorhanden, wird dieser ausgelesen und die Werte in den RDB übernommen. Wenn möglich, bleiben die Daten erhalten, wenn nötig, wird formatiert.
- Wenn weder ein RDB noch ein Evolution2.x-Bootblock vorhanden ist, so werden sinnvolle Vorgabewerte für die Festplattenparameter und die Partitionen benutzt. Eine Formatierung erfolgt dann in jedem Fall.
- Jede Festplatte erhält eine Partition. Die erste Festplatte erhält dabei die höchste Bootpriorität.
- Wenn formatiert wurde, wird eine bootfähige, allerdings nicht komplette WB 2.0 auf die Partition mit der höchsten Bootpriorität kopiert.

Wenn Sie von diesen Präferenzen abweichen wollen oder einfach mehr Kontrolle über die Installation haben möchten, dann steht die manuelle Installation zur Verfügung.

Diese ist naturgemäß sehr viel erklärungsintensiver. Deshalb soll diese Anleitung intensiv vor allem auf die manuelle Installation eingehen.

3.5 Die manuelle Installation

3.5.1 Das Fenster 'Angaben zum Gerät'

Wenn Sie "Manuelle Installation" anwählen, sucht SCSIInstall zunächst nach angeschlossenen SCSI-Laufwerken. Im Status-Fenster wird die Suche mit den Ergebnissen angezeigt. Das Fenster "Angaben zum Gerät" wird dann automatisch geöffnet, wobei das letzte gefundene Gerät angezeigt wird.

Sollten Sie ein anderes Laufwerk als das angezeigte installieren wollen, so können Sie im Menü die Option "Gerät wählen" benutzen, um auf ein beliebiges anderes Laufwerk umzuschalten.

Die Größe der Festplatte und die Aufteilung in Sektoren pro Track, Zylindern und Köpfe wird angezeigt. Sie können vorhandene RDBs auslesen und jederzeit die Vorgabe berechnen lassen. Auch Evolution2.x-Bootblöcke können explizit ausgelesen werden.

Wenn Sie nun "Installieren" wählen, wird die Festplatte mit den angezeigten Parametern installiert. Die Partitionen werden nicht verändert bzw. mit den Vorgabewerten angelegt.

Wenn Sie die Partitionierung manuell vornehmen oder kontrollieren wollen, dann fahren Sie fort mit "Partitionieren...".

3.5.2 Das Fenster 'Partitionieren...'

In diesem Fenster kann die Partitionierung der Festplatte(n) vorgenommen werden.

Sie können für jede Partition entscheiden, ob sie automatisch beim Systemstart eingebunden werden soll oder ob Sie die Einbindung später per 'Mountlist' vornehmen wollen. Auch die Bootfähigkeit der Partition kann festgelegt werden (grundsätzlich bootet die Partition, die a) bootfähig ist und b) die höchste Bootpriorität besitzt). Bootfähig können natürlich nur solche Partitionen sein, die automatisch eingebunden werden.

Im Fenster 'Partitionen' können Sie per Mausclick die Partitionen einzeln anwählen. Neue Partitionen können angelegt werden, vorhandene gelöscht werden.

Sind Partitionen vorhanden, die sich nicht mit bereits angemeldeten überschneiden, können diese direkt AmigaDOS-formatiert werden. Diese Formatierung geht sehr schnell, da lediglich der sogenannte "Root-Block" formatiert wird ("Quick"-Formatierung. Einige SCSI-Laufwerke benötigen ein sogenanntes "Pre-Format". Dies können Sie mit dem später erläuterten Befehl "LowLevel" durchführen.

Auf beliebige Partitionen, die bereits formatiert wurden, kann dann per 'WBInstall' eine Workbench kopiert werden. Bei Klick auf das Gadget werden Sie aufgefordert, eine Diskette in das interne Laufwerk df0: einzulegen, welche dann kopiert wird. Sie können eine beliebige Diskette benutzen, auch die Installationsdiskette, die dann aber in der startup-sequence das Installationsprogramm startet. Empfehlenswert ist daher die Verwendung einer Workbench-Diskette. Natürlich können Sie die Installation der Workbench auch später mit dem Install-Programm der WB 2.0 durchführen.

Wollen Sie die Partitionen noch genauer definieren, so klicken Sie auf "Weitere Einstellungen". Ansonsten führt "OK" zurück zum Fenster "Angaben zum Gerät". Alle durchgeführten Änderungen müssen dort per Klick auf "Installieren" in den Bootblock der Festplatte geschrieben werden, um nach dem nächsten Neustart aktiv zu werden.

Der 'logische Name' der Partition wird übrigens nicht angezeigt. Er kann in das dafür vorgesehene Feld vor einer Formatierung eingegeben werden.

3.5.3 Das Fenster 'Weitere Einstellungen'

Hier kann der Start- und Endzylinder der ausgewählten Partition exakt eingesehen und definiert werden. Auch die Bootpriorität und die Puffergröße (in Blocks, in der Regel hat ein Block 512 bytes Größe, die Vorgabe von 130 Blocks also ca. 65 kByte Größe) kann eingestellt werden. Unter diesem Puffer versteht man Arbeitsspeicher, der gewissermaßen der Festplatte zugewiesen wird. Wird nun der gleiche Inhalt mehr als einmal eingelesen, steht alles oder ein Teil bereits im Puffer, was die Lesegeschwindigkeit erhöht. Als Beispiel kann ein zweimaliges 'dir' bzw. Anklicken der Ikone dienen.

Der Puffer sollte möglichst nicht unter die Vorgabe gesenkt werden, da die Festplatte ansonsten langsamer wird.

Weiterhin kann hier auch das gewünschte FileSystem ausgewählt werden. Dabei gilt es zu bedenken,

Zweck: Wahl der Art des Screens

Beispiele: SCREENMODE WORKBENCH Das Installationsprogramm soll auf dem Workbench-Screen laufen.

SCREENMODE DEFAULT Das Installationsprogramm soll auf einem eigenen Screen, der dem der Workbench entspricht laufen.

Kommando: SCREENTOBACK

Schablone: Keine Parameter

Zweck: Screen des Installationsprogramms nach hinten bringen.

RC: ERROR, wenn Screen nicht auf.

Kommando: SCREENTOFRONT

Schablone: Keine Parameter

Zweck: Screen des Installationsprogramms nach vorn holen.

RC: ERROR, wenn Screen nicht offen.

Kommando: SEARCHUNITS

Schablone: Es werden allen angeschlossenen Geräte ermittelt. Dieses Kommando wird nur benötigt, wenn man mit Wechselplattenlaufwerken arbeitet und ein Medium gewechselt hat oder ein Gerät später eingeschaltet wurde.

Kommando: SECTORS *)

Schablone: Number/A/N

Zweck: Setzen der Sektorzahl des aktuellen Gerätes. Bei jeder Änderung von Köpfen, Sektoren oder Zylindern wird eine vorher definierte Partitionsliste gelöscht.

Kommando: SELECTPARTITION

Schablone: PhysicalName/A

Zweck: Die angegebene Partition wird zur aktuellen erklärt. Alle weiteren Änderungen werden an dieser durchgeführt.

RC: ERROR, Partition existiert nicht

Kommando: SETHSC

Schablone: RDB/S,EVO2/S,DEFAULT/S

Zweck: Die Werte für Köpfe, Sektoren und Zylinder auf Vorgaben setzen.

Beispiele: SETHSC RDB Köpfe, Sektoren und Zylinder werden, wenn möglich, aus einem bereits existierenden Rigid-Disk-Block auf dem aktuellen Gerät gelesen. Wurde ein RDB gefunden wird auch die zugehörige Partitionsliste gelesen.

SETHSC EVO2 Köpfe, Sektoren und Zylinder werden, wenn möglich, aus einem alten Evolution 2 Block gelesen. Außerdem werden auch die zugehörige Partitionen gesetzt.

SETHSC DEFAULT Aus der Gesamtblockzahl des aktuellen Gerätes werden möglichst gute Vorgaben berechnet. Es wird eine Partition definiert, die das Gerät voll ausnutzt.

Schablone: Name/A

Zweck: Setzen des physikalischen Namens eine neuen Partion.

Kommando: QUIT

Schablone: FORCE/S

Zweck: Beenden des Installationsprogramms

Kommando: READ Schablone: COMMAND/A

Zweck: Kommando ermöglicht das Auslesen der Einstellungen, die bei bestimmten Kommandos vorgenommen werden können. Solche Kommandos sind mit einem *) gekennzeichnet.

Beispiel: READ AUTOSAVEENV liefert ON oder OFF je nach Stellung des CheckMarks im Menü.

RC: ERROR, angegebenes Kommando existiert nicht oder es ist keine Rückgabe vorgesehen

RESULT: Rückgabe (abhängig vom angegebenen Kommando)

Kommando: REQUEST

Schablone: TEXT/A,TITLE/K,GADGETS/K

Zweck: Darstellung eines Requesters auf dem Screen des Installationsprogramms aus einem ARexx-Skript heraus.

Beispiele: REQUEST "Dies ist ein Requester" Es erscheint ein Requester, in dem der obige Text steht.

REQUEST "Frage?" TITLE "Test" GADGETS "ja—nein" Es erscheint ein Requester in dem der Text "Frage?" steht. Der Titel lautet "Test". Außerdem gibt es noch zwei Gadgets mit den Texten "ja" und "nein".

RESULT: Nummer des Gadgets '1', '2', ... , 'n-1', '0'

Kommando: SAFETYLEVEL *)

Schablone: LOW/S,MEDIUM/S,HIGH/S

Zweck: Mit diesem Kommando kann festgelegt werden, wieviele Sicherheitsabfragen erfolgen sollen.

Beispiele: SAFETYLEVEL MEDIUM Sicherheitsabfragen erfolgen nur an wichtigen Stellen.

SAFETYLEVEL HIGH Es erfolgen grundsätzlich Sicherheitsabfragen.

Kommando: SAVESETTINGS

Schablone: NAME

Zweck: Speichern der Einstellungen des Installationsprogramms

Beispiele: SAVESETTINGS ENVARC:Evo3I/Evo3I.Prefs Die Einstellungen werden im Env-Archive (ENVARC:) im Unterverzeichnis Evo3I unter dem Namen Evo3I.Prefs gespeichert.

SAVESETTINGS Es erscheint ein File-Requester in dem man den Pfad und den Dateinamen eingeben kann, unter dem die Einstellungen gespeichert werden sollen.

RC: WARN, bei Abbruch im File-Requester ERROR, Fehler beim Abspeichern

Kommando: SCREENMODE *)

Schablone: WORKBENCH/S,DEFAULT/S

daß unter Kickstart 2.x das FastFileSystem im ROM liegt und nicht extra geladen werden muß. Wenn der Start-Zylinder auf eins gesetzt ist, wäre dies sogar fatal (Überschneidung mit einer Partition). Andererseits ist das Laden eines FastFileSystems wichtig zum Betrieb der Festplatte unter OS 1.3. Bei Klick auf 'Auswahl' wird per File-Requester ein FileSystem geladen.

Per 'OK' kann zurück zum Fenster 'Partitionieren' gegangen werden. Die eventuellen Änderungen werden erst nach Klick auf 'Installieren' im Fenster 'Angaben zum Gerät' durchgeführt.

3.6 Die Menüleiste

In der Menüleiste können verschiedene weitere Dinge, die hauptsächlich das Programm betreffen, eingestellt werden. Auch der AREXX-Teil wird über die Menüleiste verwaltet.

Wer die Software zum Echtzeit-Farbvideodigitizer 'VLab' oder zur digitalen Audio-I/O-Karte 'Maestro Professional' kennt, wird sich zu Hause fühlen: Alle drei Programmoberflächen verwenden das 'MacroSystem-UserInterface', welches sich weitestgehend an die Richtlinien hält, welche das 'UserInterface-StyleGuide' von Commodore vorgibt.

Auf die einzelnen Menüs soll im folgenden näher eingegangen werden, wobei von links nach rechts vorgegangen werden soll.

3.6.1 Das Menü 'Projekt'

Dieses Menü befindet sich links in der Leiste.

Hier kann ein Gerät (nach Unit-Nummern von 0 bis 6) angewählt, ein SCSIInstall-Befehl direkt eingegeben (siehe AREXX-Teil), einige generelle Informationen zum Programm abgerufen sowie das Programm beendet werden.

3.6.2 Das Menü 'Einstellungen'

In diesem Menü können verschiedene Einstellungen vorgenommen werden, die das Programm selbst betreffen.

Der Screen-Typ kann festgelegt und umgeschaltet werden, wobei 'Kein Screen' zur Darstellung auf dem Workbench-Screen führt.

Die Sprache kann ebenfalls festgelegt werden. Hierbei wurde nach den Richtlinien der 'locale.library' verfahren, sodaß alle zur Verfügung stehenden Sprachen im Ordner 'catalogs' im Ordner des SCSIInstall-Programmes zu finden sein müssen.

Das Programm selbst verfügt über eine interne Sprache (Englisch), die auch ohne solche catalogs zur Verfügung steht. Alle anderen Sprachen werden nachgeladen. In den catalogs stehen dabei die Übersetzungen. Diese Technik ermöglicht beliebige Sprachen in einem Programm, wobei sogar plattdeutsche Versionen etc. denkbar wären.

Steht eine ausgewählte Sprache zur Verfügung, wird online, das heißt sofort, darauf umgeschaltet. Sämtlicher dargestellter Text wird geändert, die Größe der geöffneten Fenster automatisch angepaßt.

Weiterhin kann eingestellt werden, wie häufig Sicherheitsabfragen erfolgen sollen. Die Vorgabe lautet 'häufig', das heißt, vor beinahe jeder Änderung gespeicherter Werte wird eine Warnung erscheinen. Dies ist dann zu empfehlen, wenn die Gefahr des Datenverlustes weitgehend ausgeschlossen werden soll.

Geübte Anwender können auf die Sicherheitsabfragen dann ganz oder teilweise verzichten, was kurze Änderungen schneller möglich macht. Dies kann jeder für sich entscheiden.

Die Einstellungen des Programms, also Dinge wie Screen-Typ, Anordnung der Fenster oder Sprache, kann hier auch abgespeichert und geladen werden.

Wenn unter 'Einstellungen speichern' die automatische Archivierung gewählt wird, so wird jede Änderung beim Verlassen des Programms automatisch gesichert.

Beim nächsten Start kann dann auf diese Einstellungen zurückgegriffen werden. Aber auch mehrere, verschieden benannte Einstellungen können verwendet werden und per File-Requester geladen werden.

3.6.3 Das Menü 'Benutzer'

In diesem Menü werden die programmeigenen Makros sowie die AREXX-Skripte gesteuert.

4 Die Benutzung der 'HDToolBox'

Die HDToolbox ist das Commodore-eigene Installationsprogramm für Festplatten. Es ist © Commodore Amiga Inc., und daher nicht auf der Installationsdiskette zum Update zu finden. Es befindet sich aber auf jeder Install-Disk zur WB 2.x, im Tools-Verzeichnis. Viele Anwender werden es deshalb besitzen.

Die HDToolbox arbeitet unter 1.3 und unter 2.x, sie ist mit Evolution 3.0 kompatibel. Gegenüber der 2.x-Version der Evolution-3.0-Installationssoftware (Evo3Install) hat sie gewisse Schwächen (kein Auslesen des Evolution2.x-Bootblocks, nur Englisch, kein AREXX-Interface, kein AmigaDOS-Formatieren, Neustarts erforderlich etc.), das wird vielen Anwendern aber bekannt sein.

Wichtig ist, das die HDToolbox, wenn sie ohne spezielle Einstellungen gestartet wird, stets das "scsi.device" absucht. Dies ist das Commodore-Device zum Beispiel im A 3000. Commodore hat allerdings den Entwicklern eigener Controller die Auflage erteilt, andere device-Namen zu benutzen.

Das Device des Evolution heißt "evolution.device". Wenn Sie die HDToolbox also benutzen wollen, öffnen Sie eine Shell (CLI), und wechseln Sie zum Verzeichnis, in dem die HDToolbox gespeichert ist. Tippen Sie dann

```
HDToolBox evolution.device
```

Dann sucht die HDToolbox nach angeschlossenen Festplatten am Evolution-Controller.

Wenn Sie die HDToolbox über die Workbench starten wollen, so können Sie das gewünschte Device auch in das Piktogramm eintragen. Klicken Sie dazu einmal das Piktogramm an, wählen Sie dann aus der Menüleiste 'Information' und tragen in die Zeile 'Tool Types' ein:

```
scsi_device_name = evolution.device
```

Speichern Sie diese Einstellung. Dann kann die HDToolBox per Mausclick gestartet werden.

Die Bedienung der HDToolBox ist in der Anleitung zur WB 2.x ausführlich erläutert und soll deshalb hier nicht noch einmal erklärt werden.

5 Das AREXX-Interface

AREXX ist eine Programmiersprache, die als Schnittstelle zwischen den im Amiga gleichzeitig ablaufenden Tasks fungiert. So kann ein Programm gewissermaßen aus einem anderen heraus gesteuert werden.

Beispiele: LOADSETTINGS DEFAULT Es werden Standard-Einstellungen benutzt.

LOADSETTINGS FileName Die Einstellungen aus der Datei 'FileName' werden benutzt.

LOADSETTINGS Wird kein Parameter angegeben, erscheint ein File-Requester in dem man die Datei mit den Einstellungen angeben kann.

RC: WARN, bei Abbruch im File-Requester

Kommando: LOCKGUI *)

Schablone: ON/S,OFF/S

Zweck: Dieses nur über AREXX verfügbare Kommando schaltet die sonstigen Eingabemöglichkeiten (Menü, Gadgets, etc.) ein oder aus.

Beispiel: LOCKGUI ON Es werden nur noch Eingaben über AREXX akzeptiert.

Kommando: LOGICALNAME *)

Schablone: Name/A

Zweck: Setzen des für die Formatierung benutzten logischen Namen.

Kommando: MOUNT

Schablone: PhysicalName/A

Zweck: Start eines Macros. Wird das zugehörige File nicht als Makro identifiziert, wird es (je nach Typ) als AREXX-Skript, als Batch-Datei oder als Programm ausgeführt.

RC: RC des letzten Makro-Kommandos, FATAL, bei vorzeitigem Abbruch, ERROR, wenn Macro nicht existiert.

Kommando: MOUNT

Schablone: PhysicalName/A

Zweck: Einbinden der angegebenen Partition. Die Partition sollte anschließend nicht mehr verändert werden, da dann keine Formatierung mehr möglich ist.

Kommando: MOVEWINDOW

Schablone: START/S,STATUS/S,COMMAND/S,SELECTUNIT/S,PARAMETER/S, PARTITION/S,ADVANCEDOPTIONS/S,LEFTEDGE/A/N,TOPEDGE/A/N

Zweck: Positionieren des spezifizierten Fensters. Es kann immer nur ein Fenster positioniert werden!

RC: WARN, das angegebene Fenster ist nicht geöffnet, beim nächsten Öffnen werden trotzdem die neuen Werte benutzt.

Kommando: OPENWINDOW

Schablone: COMMAND/S,SELECTUNIT/S,PARAMETER/S,PARTITION/S, ADVANCEDOPTIONS/S

Zweck: Öffnen eines Fensters

Beispiel: OPENWINDOW PARAMETER Das Fenster, welches zur Eingabe der Festplattenparameter Köpfe, Sektoren und Zylinder dient, wird geöffnet.

Kommando: PHYSICALNAME *)

Schablone: Number/A/N

Zweck: Setzen der Kopfzahl des aktuellen Gerätes. Bei jeder Änderung von Köpfen, Sektoren oder Zylindern wird eine vorher definierte Partitionsliste gelöscht.

Kommando: HELP

Schablone: COMMAND/F

Zweck: Zum angegebenen Kommando einen Hilfstext ausgeben.

RC: FATAL, wenn angegebenes Kommando nicht existiert ERROR, wenn keine Hilfedatei gefunden
WARN, wenn keine Hilfe verfügbar RESULT: Hilfstext, sofern verfügbar

Kommando: INSTALL

Schablone: FORCE/S

Zweck: Alle vorgenommenen Einstellungen werden im Rigid-Disk-Block des aktuellen Gerätes abgelegt.

Kommando: LANGUAGE *)

Schablone: NAME,DEFAULT/S,INTERNAL/S

Zweck: Auch ohne Workbench 2.1 unterstützt das Installationsprogramm verschiedene Sprachen. Über dieses Kommando kann ein Wechsel der Sprache vorgenommen werden. Ist die gewählte Sprache nicht verfügbar, erfolgt eine Fehlermeldung.

Beispiele: LANGUAGE deutsch Es sollen die deutschen Programm-Texte benutzt werden.

LANGUAGE DEFAULT Die über das Locale-Preferences-Programm eingestellte Sprache soll benutzt werden. (Ab Workbench 2.1)

LANGUAGE INTERNAL Es sollen keine externen Texte benutzt werden. Dies spart etwa 25 kByte Speicher ein. Es stehen dann aber keine Help-Texte zur Verfügung.

Kommando: LOADMACRO

Schablone: NAME

Zweck: Mittels diesem Kommandos wird der Dateiname eines zu ladenden Makros festgelegt. Aus Speicher- und Geschwindigkeitsgründen erfolgt hiermit noch kein Laden. Das Makro wird bei der ersten Benutzung automatisch geladen. Die in die Makroliste aufgenommenen Makronamen werden zusammen mit den anderen Einstellungen abgespeichert. Makros sind Textdateien und müssen mit einem #EVO3I beginnen. Wird bei der Ausführung des Makros kein #EVO3I gefunden, wird versucht, die Datei als AREXX-Skript, Batch-Datei oder ausführbares Programm zu starten.

Beispiele: LOADMACRO Macros/MyMacro.vm Das Macro MyMacro.vm aus dem Verzeichnis Macros wird in die Makroliste aufgenommen.

LOADMACRO Es erscheint ein File-Requester mit dem man ein Makro aussuchen kann.

RC: WARN, bei Abbruch im File-Requester ERROR, wenn Makro nicht existiert

Kommando: LOADSETTINGS

Schablone: NAME,DEFAULT/S

Zweck: Laden der Einstellungen zum Installationsprogramm. Als da wären: CheckMarks im Menü; Fensterpositionen, Makroliste usw.

Das AREXX-Interface ist zur Installation natürlich nicht notwendig. Viele Anwender, die keine Erfahrung mit AREXX haben, werden dieses Feature von Evolution 3.0 niemals verwenden. Für AREXX-Benutzer dagegen ist ein solches Interface ein absolutes Muß! Die Implementation von AREXX ist ein wichtiger Schritt in Richtung Amiga-Multitasking-Kompatibilität.

AREXX ist übrigens ein unabhängiges Produkt, welches aber Bestandteil von AmigaOS 2.0 ist.

5.1 Kurzbeschreibung der Parameterarten

Für jeden Befehl werden die möglichen Parameter aufgelistet. Ein Kürzel hinter jedem Befehl gibt an, wann und wie die verwendeten Optionen angegeben werden müssen.

kein / Der Parameter kann, muss aber nicht angegeben werden.

/A Der Parameter muss angegeben werden.

/N Es kann eine Zahl angegeben werden.

/A/N Es muss eine Zahl angegeben werden.

/S Der Parameter ist ein Schlüsselwort als Schalter.

/M Es können ein oder mehrere Parameter angegeben werden.

/K Dem Parameter muss (wenn er angegeben wird) seinem Schlüsselwort folgen.

5.2 Rückgabewerte und Fehlercodes

Es folgen die Rückgabewerte und Fehlercodes beim Aufruf der Kommandos über AREXX.

Es existieren folgende Fehlercodes:

RC = OK = 0: alles in Ordnung
WARN = 5: Warnung, Abbruch im File-Requester etc.
ERROR = 10: Das Kommando konnte nicht korrekt ausgeführt werden
FATAL = 20: Fehlerhafte, bzw. fehlende Parameter
FAIL = -1: Kein Kommando des Installationsprogramms

Schließen sich /S-Parameter gegenseitig aus, oder muß mindestens einer angegeben werden, erhält man einen Fehlercode von 20 (FATAL). Beispiel:

AUTOUSEENV -> RC=FATAL
AUTOUSEENV ON OFF -> RC=FATAL
AUTOUSEENV ON -> RC=OK
AUTOUSEENV OFF -> RC=OK

Rückgaben erfolgen über die ARExx-Variable RESULT. Dazu muß der zugehörige Fehlercode 0 sein (RC=OK). Die jeweils möglichen Rückgaben sind bei den Kommandos angegeben. Außerdem können über das Kommando READ bei den mit einem *) gekennzeichneten Kommandos die gemachten Einstellungen ausgelesen werden.

5.3 Die einzelnen ARExx-Befehle

Kommando: ABOUT

Schablone: Keine Parameter

Zweck: Es erscheint ein Requester mit einigen Informationen zum Programm. Zusätzlich zu den üblichen Mitteilungen wie Versionsnummer und Hersteller, wird auch der Name des ARExx Ports und des Public Screens angegeben.

Wird das Installations-Programm mehrmals gestartet, hat nur das zuerst gestartete Programm einen ARExx-Port. Diese Angabe entfällt dann bei den anderen.

Kommando: ACTIVATEWINDOW

Schablone: START/S,STATUS/S,COMMAND/S,SELECTUNIT/S,PARAMETER/S, PARTITION/S,ADVANCEDOPTIONS/S

Zweck: Mit diesem Kommando kann ein Fenster aktiviert werden. Werden mehrere Fenster angegeben, wird nur das erste Schlüsselwort in der Schablone verwendet. Durch Aufruf ohne Parameter aktiviert man das Hauptfenster. Ist das zu aktivierende Fenster nicht geöffnet, wird ein Fehlercode zurückgegeben.

Beispiele: ACTIVATEWINDOW PARAMETER Das Fenster, welches zur Eingabe der Festplattenparameter Köpfe, Sektoren und Zylinder dient wird aktiviert.

ACTIVATEWINDOW Das Hauptfenster wird aktiviert (Auf der Workbench gleich ACTIVATEWINDOW START)

RC: ERROR, wenn Fenster nicht offen

Kommando: ADDPARTITION

Schablone: Keine Parameter

Zweck: Es wird eine neue Partition mit den vorher vorgenommenen Einstellungen wie PhysicalName, Start- und Endzylinder etc. in die Partitionsliste aufgenommen

RC: ERROR, kein Platz für weitere Partition, Es existiert schon eine Partition mit diesem Namen oder sonstiger Fehler.

Kommando: AREXXDEBUG *)

Schablone: ON/S,OFF/S

Zweck: Über dieses Kommando kann ein spezieller Debug-Modus für ARExx-Skripts ein- bzw. ausgeschaltet werden.

Beispiele: AREXXDEBUG ON Alle folgenden ARExx-Kommandos, sowie ihre Rückgabewerte, werden in einem Fenster ausgegeben.

AREXXDEBUG OFF Beenden des Debug-Modus.

Kommando: GETFREE

Schablone: Keine Parameter

Zweck: Ermittlung des noch nicht partitionierten Platzes des aktuellen Gerätes in KByte bzw. MByte

Kommando: GETFULLNAME

Schablone: PRESETS/S,MACROS/S,FILESYSTEM/S,USER/S

Zweck: Der Dateiname mit zugehörigen Pfad aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Dateiname mit komplettem Pfad

Kommando: GETPATH

Schablone: PRESETS/S,MACROS/S,FILESYSTEM/S,UCER/S

Zweck: Der Pfad aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Pfad

Kommando: GETPATTERN

Schablone: PRESETS/S,MACROS/S,FILESYSTEM/S,USER/S

Zweck: Der Pattern-String aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Pattern

Kommando: GETPUBSCREENNAME

Schablone: Keine Parameter

Zweck: Ermittlung des Public-Screen-Namens des Installationsprogramms. Ist kein eigener Screen geöffnet, gibt dieses Kommando den Namen 'Workbench' zurück, sonst 'EVO3L1' bzw. 'EVO3L2' etc. ARExx-Skripts sollte keine konstanten Namen, sondern immer die Resultate dieser Funktion verwendet werden.

RESULT: Name des Public-Screens

Kommando: GETUNITNUMBER

Schablone: Keine Parameter

Zweck: Über dieses Kommando kann die Anzahl der gefundenen Geräte ermittelt werden.

Kommando: GETVISITORNUMBER

Schablone: Keine Parameter

Zweck: Hiermit kann die Anzahl Fenster oder Locks auf dem Screen des Installationsprogramms festgestellt werden, welche nicht von ihm selbst stammen. Erhält man einen Wert) 0, kann dieser Screen nicht geschlossen werden. Bei QUIT oder einem Wechsel des Screen-Typs bleibt der Screen somit geöffnet! Ein Hintergrund-Task überwacht dann den Screen und Schließt ihn automatisch wenn das letzte Visitor-Window diesen Screen verlassen hat.

RESULT: Anzahl fremder Fenster oder Locks auf dem Screen des Installationsprogramms

Kommando: HEADS *)

Kommando: DOSTYPE *! Tre

Schablone:(Typm,OFS/S,FFS/S

Zweck:

Beispiel: DOSTYPE FFS Der Dos-Typ der aktuellen Partition ward auf 0x444F5301 gesetzt.

Beispiel: DOSTYPE 554F5301 Der Dos-Typ der aktuellen Partition wird auf 0x554F5309 gesetzt.

RC: FATAL, kein Parameter

Kommando: ENDCYLINDER *)

Schablone: Cylinder/A/N,FORCE/S

Zweck: Setzen des Endzylinders der aktuellen Partition. Ist diese Partition bereits in der Partitionsliste eingetragen erfolgt eine Sicherheitsabfrage. Nach einer einmaligen Bestätigung dieser Sicherheitsabfrage erfolgen bei weiteren Änderungen keine mehr.

Kommando: ERASERIGIDDISKBLOCK

Schablone: FORCE/C

Zweck: Sämtliche für einen Bigid-Disk-Block möglichen Blöcke des aktuellen Gerätes werden Überprüft und bei Vorhandensein eines solchen wird dieser gelöscht.

Kommando: FILESYSTEM *)

Schablone: Name

Zweck: Wahl eines FileSystems für die aktuelle Partition.

Beispiel: FILESYSTEM Beim Aufruf ohne Parameter erscheint ein File-Requester.

RC: WARN, Abbruch im File-Requester

Kommando: FORMAT

Schablone: PhysicalName/A Zweck: Die angebene Partition wird ins System eingebunden und anschließend formatiert (Format QUICK). Existiert bereits ein Device mit dem angegebenen Namen oder überschneidet sich die Partition mit anderen im System eingebundenen Devices, kann vom Installationsprogramm aus keine Formatierung vorgenommen werden! Die Partition kann dann erst nach dem nächsten Reset formatiert werden.

RC: ERROR, Partition konnte nicht eingebunden werden oder Fehler beim Formatieren

Kommando: GETCAPACITY

Schablone: Keine Parameter

Zweck: Ermittlung der Gesamtblockzahl des aktuellen Gerätes

Kommando: GETFILENAME

Schablone: PRESETS/S,MACROS/S,FILESYSTEM/S,USER/S

Zweck: Der Dateiname aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Dateiname

Kommando: ASLREQUEST

Schablone: TITLE/A,FILE/K,SAVE/S,DIRS/S

Zweck: Über dieses Kommandos kann ein Asl-Requester geöffnet werden. Der Text der Titelleiste muß angegeben werden. Pfad und Pattern müssen bei Bedarf vorher über SETPATH USER bzw. SETPATTERN USER angegeben werden. SAVE wählt den Requester zum Speichern. Ist DIRS angegeben, werden nur Verzeichnisse und keine Dateien angezeigt.

Beispiele: ASLREQUEST "Datei speichern:" FILE "Testdatei" SAVE

ASLREQUEST "Verzeichnis wählen:" DIRS

RC: WARN, wenn Requester mit 'Cancel' abgebrochen wurde oder kein Filename angegeben wurde.

RESULT: Der komplette Dateiname mit Pfad

Kommando: AUTOINSTALL

Schablone: FORCE/S Zweck: Automatische Installation eines Gerätes. Dieses Gerät muß vorher über das Kommando UNIT gewählt werden, sofern mehrere existieren.

RC: WARN, Sicherheitsabfrage wurde nicht bestätigt. ERROR, Fehler beim Installieren

Kommando: AUTOMOUNT *)

Schablone: ON/S,OFF/C

Zweck: Festlegung, ob die gerade bearbeitete Partition automatisch ins System eingebunden werden soll oder nicht.

Beispiele: AUTOMOUNT ON Partition soll beim Systemstart eingebunden werden.

AUTOMOUNT OFF Partition soll beim Systemstart nicht eingebunden werden. BOOT wird dadurch automatisch auf OFF gesetzt.

Kommando: AUTOSAVEENV *)

Schablone: ON/S,OFF/S

Zweck: Entscheidung ob die Environment-Variable, welche die Einstellungen des Installationsprogramms enthält, automatisch beim Beenden des Programms gespeichert werden soll. Hierbei sind nicht die Festplattendaten gemeint, sondern nur die Sprache, die Fensterpositionen etc. Wird AUTOSAVEENV aktiviert, wird automatisch auch AUTOUSEENV aktiviert.

Beispiele: AUTOSAVEENV ON Die Environment-Variable soll automatisch gespeichert werden. Das Installationsprogramm startet dadurch selbst nach einem Reset in der Konfiguration, die beim letzten Verlassen vorlag.

AUTOSAVEENV OFF Die Environment-Variable soll nicht automatisch gespeichert werden.

Kommando: AUTOUSEENV *)

Schablone: ON/S,OFF/S

Zweck: Umschaltung, ob die temporäre Environment-Variable, welche die Einstellungen des Installationsprogramms enthält, automatisch beim Beenden des Programms aktualisiert werden soll. Wenn ja, startet dieses bis zum nächsten Reset immer in der Konfiguration, die beim letzten Verlassen aktiv war. Beispiele: AUTOUSEENV OFF Die Environment-Variable soll nicht automatisch aktualisiert werden.

Kommando: BOOT *)

Schablone: ON/S,OFF/S

Zweck: Entscheidung, ob von der momentan bearbeiteten Partition beim Systemstart gebootet werden soll oder nicht.

Beispiele: BOOT OFF Von dieser Partition soll nicht gebootet werden. AUTOMOUNT wird nicht beeinflusst.

Kommando: BOOTPRI *)

Schablone: Priority/A/N

Zweck: Festlegung der Boot-Priorität der aktuellen Partition.

Beispiele:

RC: FATAL, Priority außerhalb -128 bis 127.

Kommando: BREAK

Schablone: ALL/S,C/S,D/S,E/S,F/S

Zweck: Beim nächsten eingehenden ARexx-Kommando werden dem sendenden Task die angegebenen Signalbits gesetzt. Beim Neustart von ARexx-Skripts vom Installationsprogramm aus wird die Break-Aufforderung vorher gelöscht.

Kommando: BUFFERS *)

Schablone: Number/A/N

Zweck: Anzahl Puffer festlegen.

RC: FATAL, Number \geq 1

Kommando: CHANGEASLREQ

Schablone: PRESETS/S,MACROS/S,FILESYSTEM/S,USER/S,FONT/S,SCREENMODE/S,LEFTEDGE/N,TOPEdge/N,WIDTH/N,HEIGHT/N

Zweck: Festlegung der Position und Größe der Asl-Requester. Soll ein Wert nicht verändert werden, so ist -1 anzugeben.

Beispiele: CHANGEASLREQ FILESYSTEM "-1" 120 Die Y-Position des Requesters zum Filesystem-Laden wird auf 120 gesetzt. X-Position, Breite und Höhe bleiben unverändert.

Kommando: CHANGEHELPWINDOW

Schablone: LEFTEDGE/N,TOPEdge/N,WIDTH/N,HEIGHT/N,ALT/S

Zweck: Festlegung der Position und Größe des Help-Fensters. Soll ein Wert nicht verändert werden, so ist -1 anzugeben. In der momentanen Programmversion wird das Help-Fenster beim Aufruf dieses Kommandos geschlossen.

Beispiele: CHANGEHELPWINDOW "-1" 150 "-1" 400 Die Y-Position wird auf 150 und die Breite auf 200 gesetzt. X-Position und Höhe bleiben unverändert.

CHANGEHELPWINDOW 50 200 640 200 ALT Die Alternativgröße und -position wird auf die angegebenen Werte gesetzt.

Kommando: COMMANDERRORS *)

Schablone: ON/S,OFF/S

Zweck: Entscheidung ob fehlerhafte Kommandos zu einem Fehler-Requester führen oder nicht.

Beispiel: COMMANDERRORS OFF Bei Fehlerhaften Kommandos erscheint kein Requester. Statt dessen wird in RESULT ein Fehlertext zurückgegeben.

Kommando: COPY

Schablone: FROM/A,TO/A

Zweck: Kopieren einer Workbench auf eine Festplatten-Partition.

Beispiel: COPY DF0: DH0: Es wird dazu aufgefordert eine Diskette ins Laufwerk DF0: einzulegen. Nach der Bestätigung wird diese komplett auf die Partition DH0: kopiert.

RC: ERROR, Fehler beim Kopiervorgang

Kommando: CYLINDERS *)

Schablone: Number/A/N

Zweck: Setzen der Zylinderzahl des aktuellen Geräts. Bei jeder Änderung von Köpfen, Sektoren oder Zylindern wird eine vorher definierte Partitionsliste gelöscht.

Beispiel: CYLINDERS 1000 Die Zylinderzahl wird auf 1000 gesetzt.

Kommando: DEFAULTPARTITION

Schablone: FORCE/S

Zweck: Auf dem aktuellen Gerät soll nur eine Partition existieren, die dieses voll belegt. Die Anzahl Puffer wird auf 130 und die Bootpriorität auf 0 gesetzt. AUTOMOUNT und BOOT sind eingeschaltet.

Beispiel: DEFAULTPARTITION FORCE Default-Partition ohne Sicherheitsabfrage anlegen.

RC: WARN, Sicherheitsabfrage wurde nicht bestätigt

Kommando: DELETEMACRO

Schablone: NAME/M,ALL/S,FORCE/S

Zweck: Mit diesem Kommando können Makros aus dem Speicher entfernt werden. Der Parameter FORCE unterbindet mögliche Sicherheitsabfragen. Wird der Parameter ALL angegeben, werden alle Makros aus dem Speicher entfernt. Die Makro-Dateien auf Diskette oder Festplatte werden nicht gelöscht.

Beispiele: DELETEMACRO EShell.rx Das Makro mit dem Namen EShell.rx wird aus dem Speicher entfernt.

RC: WARN, eine Sicherheitsabfrage wurde nicht bestätigt. ERROR, wenn ein Makro nicht aus dem Speicher entfernt werden konnte, weil es etwa nicht vorhanden war, oder gerade abgearbeitet wird.

Kommando: DELETEDPARTITION

Schablone: PhysicalName/M,ALL/S,FORCE/S

Zweck: Entfernen einzelner Partitionen oder Löschen der gesamten Partitionsliste.

Beispiel: DELETEDPARTITION DH0: FORCE Die Partition DH0: wird ohne Sicherheitsabfrage aus der Partitionsliste entfernt.

RC: ERROR, Partition existiert nicht WARN, Sicherheitsabfrage wurde nicht bestätigt