

# 9 Arbeitshilfen

## Kurz gefasst:

- Hier finden Sie diejenigen generellen Tipps, Tricks und Informationen, die Sie wahrscheinlich bei fast jeder Director-Produktion benötigen werden.
- Sehen Sie in den Abschnitten ARBEITSTECHNIKEN → 513 und TOOLSKRIPTE UND MIAW-TOOLS → 519, wie Sie produktiver mit Director arbeiten und die Arbeitsumgebung mit eigenen Werkzeugen erweitern können.

- Erforschen Sie detailliert die Optionen bei der Projektorerstellung → 536 und die INI-Dateien → 539.
- Häufige Fehlerquellen und Probleme bei der Arbeit in Director finden Sie konzentriert in einem eigenen Abschnitt → 547.

## 9.1 Legalize it!

### Macromedia End User License Agreement (EULA)

Aus unerfindlichem Grund existiert keine Datei im Director-Verzeichnis, die das Lizenzabkommen enthält. Lediglich beim Start des Installers wird die Lizenz angezeigt und kann als Datei gesichert werden.

Der folgende Auszug ist dem Lizenzabkommen für Director 8 entnommen (vom 14.2.2000). Es sind die beiden Stellen, die die Rechte für die Distribution (von Projektoren) beschreiben, und insbesondere ausschließen, dass ein Produkt, das beliebige andere Filme darstellen kann (also eine Konkurrenz zur Macromedia Shockmachine), vertrieben wird.

ENDBENUTZER-LIZENZVERTRAG FÜR  
MACROMEDIA(r) DIRECTOR(r)- UND  
MACROMEDIA(r) SHOCKWAVE(r)-MULTI-

USER-SERVER-SOFTWARE

(...)

2. Lizenzgewährung.

(a) Nur für die Director-Software: (...)

(ii) Sie dürfen Kopien des Endabnehmer-Produkts und der dazugehörigen Macromedia Run-Time erstellen und diese Kopien zu Ihrem eigenen Gebrauch oder an Ihre eigenen Arbeitnehmer oder an eine dritte Partei ausschließlich für deren internen Gebrauch verteilen. Endbenutzer-Produkte, deren einziger Zweck das Abspielen von Animationen, Präsentationen, interaktivem Multimedia-Material, interaktiven Unterhaltungsprodukten und ähnlichen Materialien ist, dürfen nicht vertrieben werden. Endanwender-Produkte dieser Art bedürfen einer speziellen Lizenzgewährung durch Macromedia.

3. Run-Time-Verteilungsvereinbarung  
("Made With Macromedia-Programm")

(a) Falls Sie das Endabnehmer-Produkt und die dazugehörige Macromedia Run-Time außer wie in Absatz 2(a) (ii) angeführt verteilen möchten, müssen Sie einen Vertriebslizenzvertrag für Macromedia Run-Time abschließen. Weitere Einzelheiten erhalten Sie bei Macromedia. (In den USA wählen Sie die Nummer 800-449-3329. Sie finden Informationen zum "Made With Macromedia"-Programm auch auf der Website von Macromedia: [www.macromedia.com/support/programs/mwm/](http://www.macromedia.com/support/programs/mwm/).)

(b) Endabnehmer-Produkte der Lehrversion und der nicht zum Wiederverkauf überlassenen Version dürfen nicht kommerziell vertrieben werden. (...)

13. Allgemeine Bestimmungen

Dieser Vertrag unterliegt dem Recht des US-Bundesstaates Kalifornien. (...)

Macromedia, Director und Shockwave sind Marken oder eingetragene Marken von Macromedia, Inc. in den USA und/oder anderen Ländern.

### Projektoren für beide Plattformen

Muss ich Director für beide Plattformen besitzen, um Projektoren für Mac und Windows zu erstellen bzw. zu nutzen?

Praktisch gesehen: ja. Rechtlich gesehen: wahrscheinlich nein.

Director für Windows erzeugt nur Projektoren für Windows, und Director für den Mac tut dies nur für den Mac. Mit einem Universal-Projektor (vgl. Abschnitt *Projektoren* → 536) für Win-

dows, den Sie von einem anderen Entwickler bekommen, können Sie – theoretisch – auch einen Windows-Titel auf dem Mac produzieren, ohne über einen PC zu verfügen und umgekehrt. Für eine professionelle Produktion mit frühzeitigem und regelmäßigem Testing ist dieses Verfahren allerdings nicht zu empfehlen.

Offenbar toleriert Macromedia aber den Sachverhalt, dass Universalprojektoren sogar über das Internet vertrieben werden. Unter der URL <http://www.astudios.com/stubs.asp> finden Sie – für Director 7 – so genannte *Stub-Projektoren* zum Download.

### Weitere Lizenzen

Die folgenden häufig benutzten Software-Produkte müssen lizenziert werden, wenn Sie sie auf Ihrer CD-ROM vertreiben wollen:

■ Shockwave Standalone-Installer (Shockwave ohne Internet-Zugang installieren): gratis, aber ein Lizenzabkommen ist nötig (<http://www.macromedia.com/support/shockwave/info/licensing/>).

■ QuickTime: gratis oder kostenpflichtig, ein Lizenzabkommen ist nötig: <http://www.apple.com/quicktime/> bzw. <http://developer.apple.com/mkt/swl> (vgl. Kapitel 4 VIDEO UND SOUND → 265)

■ DirectX: gratis, das End User Licensing Agreement von Microsoft muss auf der CD vorhanden sein (<http://www.microsoft.com/directx/resources/>).

■ Xtras: je nach Produkt, vgl. das Produkt-Listing in Kapitel 10 XTRAS → 557.

■ Microsoft Internet Explorer (IE ActiveX Control): gratis (<http://ieak.microsoft.com/>)

■ Netscape Navigator: gratis (<http://home.netscape.com/partners/distribution/>)

Da sich entsprechende Verpflichtungen und die genauen Bestimmungen häufig ändern, sollten Sie direkt online recherchieren, welche Verpflichtungen Sie mit der Distribution von Systemtechnologien oder anderen Softwareprodukten auf Ihrer CD-ROM eingehen.

## 9.2 Arbeitstechniken

Häufig sind es die unscheinbaren Probleme bei der Handhabung eines Programmes, die Zeit und Nerven kosten. Hier finden Sie die Antworten auf häufige Fragen zum produktiven Arbeiten in Director sowie einige oft benötigte Skriptlösungen.

### Effektives Authoring

#### Arbeiten mit mehreren Director-Kopien

Im Unterschied zu vielen anderen Anwendungsprogrammen kann Director bekanntlich immer nur ein Film-Dokument öffnen, was z.B. beim Kopieren von Daten von einem Film zu anderen etwas hinderlich ist.

Auf dem Umweg, mit mehreren Instanzen oder Kopien des Programms zu arbeiten, ist jedoch auch die gleichzeitige Arbeit mit mehreren Filmen auf einem System möglich, wobei allerdings ein paar plattformspezifische Unterschiede zu berücksichtigen sind.

Windows lässt prinzipiell den mehrfachen Start einer Anwendung zu. Mit jedem Doppelklick auf ein Director-Dokument wird – gelegentlich auch unbeabsichtigt – eine neue Instanz des Programms gestartet.

Der Versuch, Daten von einer Instanz zur anderen zu kopieren, misslingt allerdings. Denn Director kopiert nur einen Verweis auf die Daten in die Zwischenablage. Beim Einfügen versucht dann die aktive Instanz des Programms, das Dokument, aus dem kopiert werden soll, erneut zu öffnen. Dies erzeugt einen Lesefehler, solange das Dokument noch mit der inaktiven Instanz geöffnet ist. Sie kommen also nicht umhin, entweder nur das Dokument (durch Erzeugen eines neuen Films) oder die gesamte Programminstanz zu schließen, bevor Sie die Zwischenablage einfügen.

Anders unter dem MacOS: Hier wird eine Applikation nur einmal gestartet. Ein Doppelklick auf ein Director-Dokument bei aktivem Programm schließt zunächst das aktuelle Doku-

ment und lädt das zu öffnende. Sie können jedoch eine Kopie des gesamten Director-Programmordners erstellen (was auf ein und demselben System keine lizenzrechtlichen Probleme bereitet) und diese Kopie parallel betreiben. Dokumente müssen allerdings entweder durch Drag&Drop auf das Programm-Icon oder über das DATEI-Menü geöffnet werden, da ein Doppelklick immer die in der Schreibtischdatei registrierte Programmdatei aktiviert. Zwischen zwei Programminstanzen ist das Kopieren von Daten problemlos möglich. Um Verwechslungen der Instanzen zu vermeiden, ändern Sie am besten den Namen der kopierten Programmdatei. Innerhalb von Director lässt sich der Programm-Ordner im Zweifelsfall mit der Anweisung

```
put the applicationPath
```

im Nachrichtenfenster anzeigen.

#### Sprites kopieren: schwerwiegender Bug

In der Liste bekannter Fehler für Director 8.0 ([http://www.macromedia.com/support/director/release-notes/8/releasenotes\\_8.htm#known](http://www.macromedia.com/support/director/release-notes/8/releasenotes_8.htm#known)) findet sich in der Rubrik „Score“ folgender Hinweis:

*When copying sprites from one movie that has external casts to another (that also has external casts), the casts are appended to the destination movie, but the sprites pasted still reference their original castLibNum.*

Dies bedeutet schlicht, dass das Kopieren von Drehbuchinhalten immer dann fehlschlägt, wenn sowohl der Quell- als auch der Zielfilm externe Besetzungen verwenden und die Nummern einander entsprechender Besetzungen nicht zufällig gleich sind.

Konkret ist das Ergebnis des Einfügens ein Drehbuch, dessen Darstellerreferenzen (inklusive zugewiesener Behaviors) fehlerhaft sind, da sie auf eine andere Besetzung verweisen. In Einzelfällen mag eine Reparatur des Drehbuchs mit

Hilfe des Befehls DARSTELLER TAUSCHEN sinnvoll sein.

Um einen Datenverlust durch diesen Bug zu vermeiden, müssten die Besetzungen beider Filme die gleiche Nummer besitzen. Die Reihenfolge der Besetzungen eines Films lässt sich jedoch nachträglich nicht ändern. Wenn Sie also Drehbuchinhalte aus Filmen mit externen Besetzungen kopieren müssen, besteht der Workaround darin, die Darsteller aller in der Sequenz verwendeten Sprites temporär in eine interne Besetzung zu verschieben, zu speichern und erst dann die Sequenz in die Zwischenablage zu kopieren.

### Zwei Director-Filme verbinden

Um zwei Director-Filme zu einem zu verbinden, gehen Sie wie folgt vor:

1. Importieren Sie mit DATEI/IMPORTIEREN... einen Film in den anderen. Wählen Sie dabei „Standard-Import“, so dass alle Darsteller importiert werden und die Drehbuchinformation in einen Filmloop-Darsteller geschrieben wird.
2. Wählen Sie den Filmloop-Darsteller in der Besetzung aus, und kopieren Sie ihn in die Zwischenablage.
3. Wählen Sie eine Zelle an der Stelle im Drehbuch aus, wo der Film eingefügt werden soll, und fügen Sie den Inhalt der Zwischenablage dort ein.

Damit werden alle Informationen des eingefügten Films (insbesondere auch die der oberen Kanäle (Tempo, Bildübergänge etc.) wiederhergestellt.

### Filme updaten, schützen oder als Shockwave-Film sichern

Folgende Befehle stehen für die Bearbeitung von Director-Filmen zur Verfügung. Alle Befehle im Menü XTRAS/FILME AKTUALISIEREN... können auf eine Liste von Director-Filmen und -Besetzungen angewendet werden.



Abbildung 9.1: Dialogfeld Xtras/Filme aktualisieren

■ **AKTUALISIEREN** (Menü XTRAS/FILME AKTUALISIEREN...) und **DATEI/KOMPAKT SICHERN**

Schreibt die Arbeitsdatei neu. Dabei werden Dateireferenzen überprüft und aktualisiert, die Datei wird kompakter, da Zwischenzustände, die Director beim normalen Sichern nicht entfernt hat, nun gelöscht werden. Die neue Datei ist nicht geschützt.

■ **SCHÜTZEN** (Menü XTRAS/FILME AKTUALISIEREN...)

Die Datei wird neu geschrieben, Scripttexte werden entfernt, die Datei wird als „geschützt“ markiert, kann also mit Director nicht mehr direkt als Film oder Besetzung geöffnet werden. Der geschützte Film hat die Endung .dxx, eine Besetzung erhält die Extension .cxt.

■ **IN SHOCKWAVE-FILM(E) KONVERTIEREN** (Menü XTRAS/FILME AKTUALISIEREN...) und die Optionen unter **DATEI/VERÖFFENTLICHEN** bzw. **DATEI/VERÖFFENTLICHUNGSEINSTELLUNGEN**.

Wie **SCHÜTZEN**, zusätzlich wird die eingestellte Komprimierung für Audio- und Grafikdarsteller angewendet, und die zusätzlich abgelegten Daten für importierte GIFs und JPEGs werden entfernt. In den Veröffentlichungseinstellungen haben Sie – für einen einzelnen Director-Film – weitere web- bzw. browserspezifische Optionen. **DER KONVERTIERTE FILM HAT DIE ENDUNG .DCR, EINE BESETZUNG ERHÄLT DIE EXTENSION .CCT**.

Wirklich geschützt sind sowohl bei .DXX/.CXT als auch bei .DCR/.CCT- (Shockwave-) Dateien nur die Skripte, da beim Schützen der Skripttext entfernt wird und nur die kompilierte Form übrig bleibt. Die anderen Medien sind nicht geschützt, da sie von geeigneten Tools aus dem entsprechenden Film herauskopiert werden können.



„alle Dateien“ und nicht nur Textdateien zu öffnen. Skripttexte können Sie zumeist direkt aus dem Dokument kopieren und wieder in Director nutzen.

### MacOS: Formularblock

Am MacOS können beliebige Dokumente als Formularblock genutzt werden. Es reicht aus, einen Director-Film, der die gewünschten Einstellungen (wie z.B. die Bühnengröße und Farbe, die Einstellungen im Menü MODIFIZIEREN/FILM/XTRAS, ja selbst die Position der Bühne oder Standardskripte) enthält, zu sichern und dann im Finder als Formularblock auszuzeichnen.



Abbildung 9.3: Die Option „Formularblock“ macht aus einem Director-Film ein Standarddokument.

### MacOS: Alte Alias-Dateien entfernen

Ist die Startzeit Ihrer Director-Anwendung zu lang, so suchen Sie im Xtras-Ordner nach Alias-Dateien, deren Zieldatei verloren gegangen ist. Ein einziges „totes“ Alias hat die Startzeit von Director von 7 auf 19 Sekunden verlängert.

### Dateinamen

Director ergänzt wenn nötig selbstständig Dateierweiterungen. Mit der Angabe

```
member(m).fileName = "@/media/hans"
```

würde also auch die Datei HANS.MOV oder HANS.JPG gefunden. Wenn mehrere Dateien mit derselben Bezeichnung (ohne Dateierweiterung) vorhanden sind, ist das Ergebnis nicht voraussehbar. Vermeiden Sie, Verzeichnisse so zu benennen wie Dateien im gleichen Verzeichnis; wenn Sie in einem solchen Fall nämlich die obige Methode anwenden, so erhalten Sie eine Fehlermeldung.

### Spielstände oder andere Zustände einer Anwendung speichern

Die drei Teile dieser Frage sind:

- Wie speichere ich Text- oder andere Informationen aus einer Director-Anwendung heraus?
- Wer soll etwas mit den Daten anfangen? Sollen sie (z.B. auf einen Webserver zur Analyse) weitergegeben werden, oder dienen sie nur dazu, den (lokalen) Zustand zu sichern?
- Welches sind die entscheidenden Informationen, die es mir erlauben, einen Zustand meiner Anwendung wiederherzustellen? Welche Informationen muss ich denn überhaupt speichern?

Aus der folgenden Tabelle können Sie für die ersten beiden Fragen die passenden Möglichkeiten ermitteln:

| Methode                                  | geeignet für   | siehe                      |
|--|--|----------------------------|
| Lingo:<br>setPref(), getPref()           | Projektor auf Festplatte;<br>Shockwave (ohne Rückmeldung zum Server)   | Kapitel 7, SHOCKWAVE → 413 |
| FileIO-Xtra:<br>Text schreiben und lesen | Projektor (Festplatte oder CD-ROM)   | Kapitel 5, TEXT → 302      |
| Netlingo:<br>getNetText(), postNetText() | Shockwave, Projektor mit Internet-Anbindung;<br>für die Weitergabe von Daten;<br>erfordert CGI auf dem Webserver | Kapitel 7, SHOCKWAVE → 387 |

Tabelle 9.1: Möglichkeiten zur Speicherung von externen Spielständen

Der letzte Punkt ist schwieriger zu fassen. Möglich wäre es beispielsweise, alle globalen Variablen in einen Textstring zu verwandeln und zu speichern:

```
-- Aufruf: setpref("myPrefs",
string(getglobals()))

on getglobals
  temp = the globals
  retval = []
  repeat with i = 1 to temp.count()
    set gl_name = temp.getPropAt(i)
    set gl_val = temp[gl_name]
    retval[gl_name] = gl_val
  end repeat
  return retval
end
```

Listing 9.1: Globale Variablen speichern (Filmskript)

Beim Programmstart würde die Datei eingelesen und die globalen Variablen würden wiederhergestellt:

```
on setglobals
  aList = list(getPref("myPrefs"))
  repeat with i = 1 to aList.count()
    (the globals)[aList.getPropAt(i)] = aList[i]
  end repeat
end
```

Listing 9.2: Globale Variablen einlesen und setzen (Filmskript)

Auch wenn Sie globale Variablen für alle Informationen angelegt haben, die für Sie wichtig sind (z.B. auch the frame oder Zustände von Eingabefeldern und Buttons), würde diese Methode einiges an Aufwand bedeuten, denn Sie müssten dann auch alle Zustände entsprechend restaurieren (Abspielkopf positionieren, Textfelder und Button-Zustände setzen etc.)

Sinnvoller ist es, die Anwendung so aufzubauen, dass mit wenigen gespeicherten Informationen und einem Handler, der Spielstände, Quizergebnisse oder die Liste der besuchten Seiten wiederherstellt, die gewünschte Situation erreicht werden kann. Das wird allerdings für jede konkrete Anwendung anders aussehen.

## Alle Quit-Tastenkombinationen abfangen

Das folgende Skript nutzt die System-Property the exitLock, um das Beenden eines Projektors mit den üblichen Tasten(-kombinationen) zu unterbinden. Außerdem springt der Film, falls [Esc], [Alt] [F4], [Strg]/[⌘] [Q] oder [Strg]/[⌘] [.] gedrückt wurden, auf die Drehbuchmarkierung „Ende“. „Harte“ Methoden, eine Applikation zu beenden (am MacOS beispielsweise [⌘] [Esc], Windows [Strg] [Entf]), können Sie allerdings nicht abfangen.

```
on startMovie
  the exitlock = TRUE
end startMovie
```

```

on keydown
  kc = the keyCode
  qflag = 0
  if kc = 53 OR (the optionDown AND kc = 118)
    -- Escape (MacOS/Win), Alt + F4 (Win)
    qflag = 1
  else
    if the commandDown then
      -- "Q" oder "."
      if kc = 12 OR kc = 47 then qflag = 1
      --
    end if
  end if
  if qflag then go "Ende"
end

```

*Listing 9.3: Exitlock „de Luxe“ (Filmskript)*

Beachten Sie, dass the `exitlock` auch Auswirkungen auf die Funktion des Schließfeldes und (unter Windows) auf die Funktion des SCHLIESSEN-Befehls in der linken Ecke des Projektorfensters hat, die beide sichtbar sind, wenn der Projektor nicht im Vollbildmodus abgespielt wird. Bei the `exitlock = 0` funktioniert das Schließen des Fensters normal, bei the `exitlock = 1` wird das Schließen auch auf diesem Weg unterbunden.

## Geschützten Code erstellen

Skripte werden durch die Umwandlung eines Director-Filmes in einen geschützten Film oder Shockwave-Film (bzw. eine geschützte Besetzung oder Shockwave-Besetzung) wirkungsvoll geschützt. Damit können sie allerdings auch nicht mehr in anderen Projekten genutzt werden, so dass dies kein Weg ist, um geschützten Code zu distribuieren.

In einer schon etwas älteren Technote stellt Macromedia eine Lösung vor, wie Movies-in-a-Window als geschützte Code-Container genutzt werden können (<http://www.macromedia.com/go/8140>). In Director 8 haben Sie mindestens zwei weitere Möglichkeiten, die mir doch etwas eleganter erscheinen:

- die Einbindung von geschützten Besetzungen
- die Einbindung eines geschützten Films als Sprite

## Geschützte Besetzung

Eine geschützte Besetzung, die statt der normalen Dateieindung `.CXT` wieder die Endung `.CST` erhält, kann weiterhin in Director-Projekten genutzt werden. Enthaltene Behaviors können Sie auf Sprites ziehen etc. Allerdings gibt es bei diesem Verfahren drei Probleme:

- Sie können die Fehlermeldung erhalten, dass die eingebundene Besetzung geschützt ist und nicht geöffnet werden kann. Dies ist entweder unerheblich (da sie dennoch nutzbar ist), oder es weist auf das folgende Problem hin.
- Zuweilen geht die Referenz auf die externe Besetzung verloren. Das lässt sich beheben, indem Sie unter MODIFIZIEREN/FILM/BESETZUNGEN die Castlib auswählen und mit EIGENSCHAFTEN einen neuen Pfad zuweisen.
- Beim Ausführen des Befehls ALLE SKRIPTE REKOMPIlierEN und beim Öffnen von Skriptdarstellern aus der geschützten Besetzung werden die Skripte zerstört (da versucht wird, den leeren Skripttext zu rekompilieren). Dies ist die Schwachstelle dieser Methode.

## Geschützten Film als Sprite verwenden

Wenn Sie Ihre Skripte als Film- und/oder Parent-Skripte anlegen können und keine direkt einsetzbaren Behaviors benötigen, ist dieser Ansatz zuverlässig nutzbar. Die Grundidee ist, dass ein als Sprite eingebundener Director-Film ebenfalls die normalen Events erhält (z.B. `startMovie`, `exitFrame` etc.) und auch als Skriptcontainer fungieren kann.

Platzieren Sie das folgende Skript in einen leeren Film, sichern Sie diesen, und importieren Sie ihn als externen, gelinkten Darsteller (Typ: `#movie`). Schalten Sie im Property-Inspektor die Eigenschaft `SKRIPTS AKTIVIEREN` an. Ziehen Sie den

Darsteller dann in das Drehbuch, und starten Sie die Wiedergabe.

```
on startMovie
  put "Eingebundener Film startet"
end

on exitFrame
  put "Eingebundener Film führt regelmäßig \
  Befehle aus."
end
```

*Listing 9.4: Eventhandling Movie-in-a-Sprite (Filmskript)*



Beispiel auf CD-ROM im Verzeichnis  
WORKSHOP/KAPITEL09/PROTECTCODE

Ebenso könnte auf `startMovie` des eingebundenen Films natürlich auch die Instanz eines Parent-Skripts gebildet und in eine globale Variable geschrieben werden, so dass die Funktionen des Parent-Skripts dann für den Host-Film zur Verfügung stehen. Auf der CD-ROM finden Sie ein entsprechendes Beispiel, in dem die JPEG-Export-Funktionalität in einem geschützten Film untergebracht ist.

Sie können auch Filmskripte des eingebundenen Films aufrufen. Dieses Skript im Movie-in-a-Sprite

```
on hallo
  return "Hallo Welt"
end
```

rufen Sie im Host-Film so auf:

```
tell sprite(1) to put hallo()
-- "Hallo Welt"
```

*Listing 9.5: Filmskript im Movie-in-a-Sprite aufrufen (Nachrichtenfenster)*

Diese Funktionalität ist unabhängig davon, ob der eingebundene Film geschützt ist oder nicht.

## 9.3 Tool-Skripte und Movies-in-a-Window

Wiederkehrende Arbeiten gibt es bei jedem Projekt in Director genug: eine große Anzahl von Textdarstellern formatieren, immer die gleichen Zeilen als Grundgerüst eines Behaviors tippen oder auch nur wiederholt im Nachrichtenfenster ähnliche Aufrufe eingeben – wenn Sie mehr als eine Viertelstunde mit der gleichen stupiden Arbeit verbringen, lohnt es sich darüber nachzudenken, ein eigenes kleines Tool in Lingo zu programmieren.

In meinem aktuellen Director-Ordner befinden sich unter anderem folgende Tools (teilweise selbst programmiert, teilweise von anderen Anwendern); manche sind über Eingabefelder konfigurierbar, andere bieten nur einen Button und führen eine Aktion aus (Abbildungen 9.4 bis 9.12).

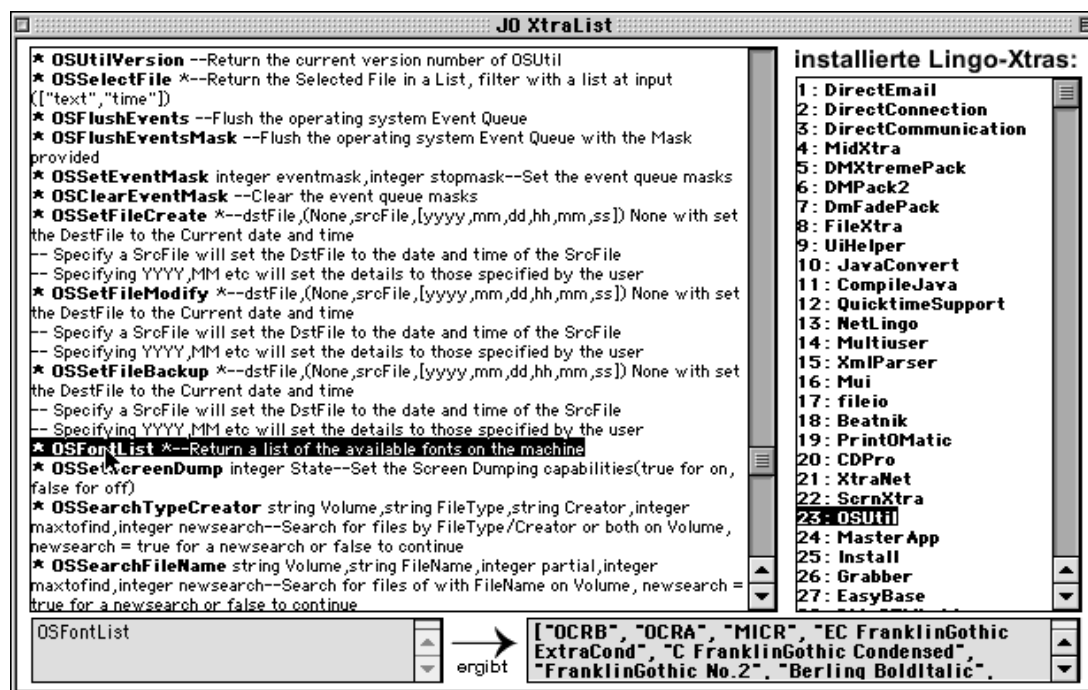


Abbildung 9.4: XtraList: Informationen über alle Lingo-Xtras anzeigen

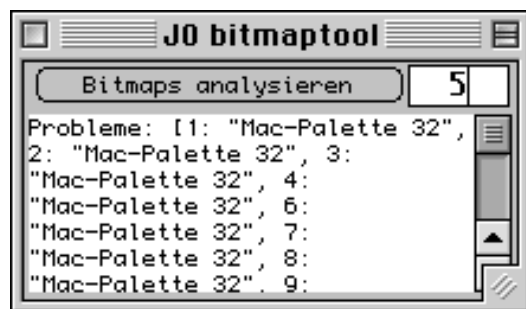


Abbildung 9.5: Bitmap-Farbtiefe analysieren

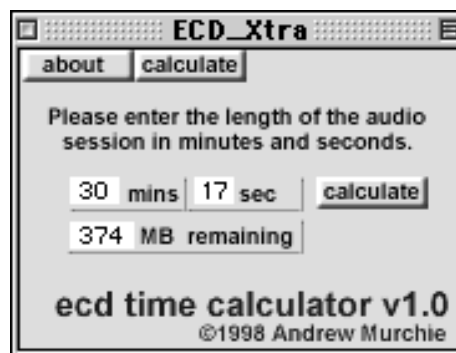


Abbildung 9.7: Verbleibende Datenmenge bei Enhanced CDs (Audio-CD und Daten)

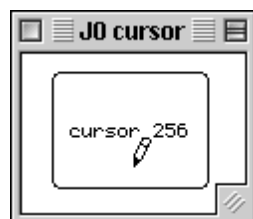


Abbildung 9.6: Gedächtnisstütze für Cursor-Formen am Mac

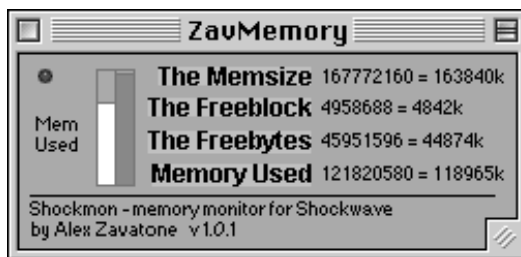


Abbildung 9.8: Memory Watcher für Shockwave



Abbildung 9.9: Pfadanimation eines Sprites zur späteren Verwendung in eine Liste schreiben



Abbildung 9.10: Hilfe beim Erstellen von Behaviors



Abbildung 9.11: Mausinformationen



Abbildung 9.12: Kleines Navigations-Tool mit Anzeige des Markernamens

Außerdem gibt es in meinem XTRAS-Ordner einen Unterordner MAKROS, wo ich einige selbst erstellte Director-Makros untergebracht habe. Auch sie sind Movies-in-a-Window, die allerdings nicht konfiguriert werden müssen und so ohne „Interface“ auskommen.

Movies-in-a-Window (MiaWs), die im XTRAS-Ordner von Director liegen, werden automatisch im Menü XTRAS der Authoring-Umgebung angezeigt. Liegen sie in einem Unterordner des XTRAS-Ordners, so ist der Name des Unterordners gleichzeitig der Menüeintrag im XTRAS-Menü. Wählt man den Menübefehl aus, so wird der Director-Film als Movie-in-a-Window geöffnet.

Wollen Sie MiaW-Tools (oder MiaWs allgemein) außerhalb der Authoring-Umgebung und ohne die Hilfe des XTRAS-Menüs verwenden, so

öffnen Sie das Fenster mit dem Befehl `open window windownameOderPfad` mit Lingo. MiaWs funktionieren auch im Projektor. In Shockwave können MiaWs nicht eingesetzt werden, dort ist es aber möglich, mehrere Filme in verschiedenen Fenstern oder Frames des Browsers zu öffnen.

### Movies-in-a-Window

Für den Objekttyp `window` verfügt Lingo über einige System-Properties und Funktionen:

- `the windowList`: enthält Referenzen auf alle offenen Fenster außer der Bühne
- `the activeWindow`: enthält eine Referenz auf das aktive Fenster, was auch die Bühne sein kann; es muss nicht unbedingt das vorderste Fenster sein

(Bühne kann aktiv sein, ohne das vorderste Fenster zu sein)

- `the frontWindow`: enthält eine Referenz auf das vorderste Fenster, was auch die Bühne sein kann
- `windowPresent("einFensterName")`: liefert TRUE zurück, wenn das Fenster existiert

Außerdem gibt es eine Reihe von Befehlen, um Fenster zu öffnen, zu schließen und zu bewegen:

- `FensterReferenz.open()`: öffnet das Fenster bzw. macht es wieder sichtbar
- `FensterReferenz.close()`: versteckt das Fenster, ohne es zu löschen
- `FensterReferenz.forget()`: löscht das Fenster aus dem Speicher und der `windowList`
- `FensterReferenz.moveToBack()`: bewegt das Fenster nach hinten
- `FensterReferenz.moveToFront()`: bewegt das Fenster nach vorn

In Director 7 gab es ein Problem mit dem Lingo-Befehl `forget()`. Macromedia empfahl in Technote 14069 (<http://www.macromedia.com/go/14069>) auf `forget()` zu verzichten.

Folgende Properties können für Fenster aus-gelesen bzw. gesetzt werden:

- `FensterReferenz.sourceRect`: beschreibt die Originalgröße und -position des Fensters in Bildschirmkoordinaten (entspricht den Ursprungsdaten des Filmes, der als `MiaW` geöffnet wird)
- `FensterReferenz.rect`: wenn das `rect` kleiner ist als das `sourceRect`, so wird der dargestellte Bereich des Films abgeschnitten (Cropping)
- `FensterReferenz.drawRect`: beschreibt die Darstellungsgröße des Director-Films innerhalb des `rect` des Fensters. Die meisten Darstellertypen werden skaliert, wenn Sie das `drawRect` ändern. Die „alten“ Darsteller `Feld`, `Button`, „Markierungsfeld“ und „Auswahlfeld“ verhalten sich beim verändern dieser fenstereigenschaft nicht immer wie gewünscht.

Die Eigenschaft `rect` kann nicht ganz beliebig gesetzt werden. So besteht eine Mindestbreite von 70 Pixeln (Authoring) und 140 Pixeln (Projektor). Soll die Bühne den ganzen Bildschirm ausfüllen, so müssen Sie `rect` und `drawRect` entsprechend setzen. Das folgende Skript nimmt den ersten Eintrag aus der `desktopRectList` (dies ist der Systemmonitor, auch bei einem Setup mit mehreren Monitoren) und ändert `drawRect` und `rect`:

```
(the stage).drawRect = the desktopRectList[1]
(the stage).rect = the desktopRectList[1]
```

- `FensterReferenz.fileName`: Pfad und Dateiname des Filmes im Fenster
- `FensterReferenz.modal`: wenn TRUE, kann das Fenster durch einen Klick (z.B. auf die Bühne) nicht deaktiviert werden; am MacOS gilt die Modaleinstellung systemweit
- `FensterReferenz.title`: erlaubt das Setzen des Fenstertitels
- `FensterReferenz.titleVisible`: wenn TRUE, wird die Titelleiste des Fensters angezeigt
- `FensterReferenz.windowType`: bezüglich einzelner Fenstertypen vgl. Kapitel 8, PLATTFORMSPEZIFISCHE ARBEITSTECHNIKEN → 470

Diese Event-Handler können auf Events reagieren, die Director automatisch bei Statusänderungen des Fensters erzeugt:

- `on activateWindow`: das Fenster wurde aktiviert (z.B. per Klick)
- `on closeWindow`: das Fenster wurde geschlossen (versteckt)
- `on deactivateWindow`: ein anderes Fenster wurde aktiviert
- `on moveWindow`: das Fenster wurde bewegt (Ziehen mit der Maus)
- `on openWindow`: das Fenster wurde geöffnet
- `on resizeWindow`: die Resize-Ecke am Fenster wurde benutzt
- `on zoomWindow`: die Zoom-Box am Fenster wurde betätigt

Eine Fensterreferenz kann in der Form (`window "Hans"`), `window ("Hans")`, `window ("@/Unterordner/Hans.dir")` oder im Falle der Bühne als (`the stage`) angegeben werden. Selbstverständlich sind auch andere Pfadschreibweisen als die mit `@` möglich.

Neben diesen MiaW-spezifischen Lingo-Elementen können Sie natürlich auch alle anderen Lingo-Befehle, Properties etc. in einem Movie-in-a-Window nutzen.

### Besseres Update beim Ziehen von Fenstern

Wenn Sie MiaWs implementieren, die ohne eigene Titelleiste auskommen (Fenstertypen 1, 2 und 3), diese Fenster aber dennoch bewegen wollen, so sollten Sie sich die einfache Lösung mit `repeat while the stillDown` verkneifen: Sie sorgt nämlich dafür, dass das Bild der Bühne hinter dem MiaW nicht aufgefrischt wird, so dass hässliche Spuren des MiaWs entstehen.

Statt

```
on mouseDown me
  repeat while the stillDown
    -- Fenster bewegen
  end repeat
end
```

sollten Sie also folgende Lösung nutzen (z.B. als Behavior auf einer selbst gestalteten Fenstertitleiste):

```
on mouseDown me
  (the actorList).add(me)
end

on stepFrame me
  if the stillDown then
    -- Fenster bewegen
  else
    (the actorList).deleteOne(me)
  end if
end
```

*Listing 9.6: MiaW bewegen (Behavior-Skript)*

In der Zeile `-- Fenster bewegen` würden Sie `window(w).rect` so verändern wie benötigt. Der Vorteil dieser Lösung ist, dass sie dem Betriebssystem die Gelegenheit gibt, die Bühne aufzufrischen. Bei schwacher Grafikleistung des Systems wird dennoch ein Trails-Effekt sichtbar sein.

## Prinzipielles zu MiaW-Tools

### Adressierungen in Director

Um ein Tool-Xtra als Director-Film zu realisieren, ist es häufig nötig, ein Objekt (das MiaW) eine Aktion im anderen Objekt (dem offenen Hauptfilm, `the stage`) ausführen zu lassen. Objektreferenzen (wie beispielsweise `member(1,1)`) können aber sowohl im MiaW als auch im Hauptfilm auftreten. Andere Objektreferenzen (z.B. `the scoreSelection`) sind nur auf die Authoring-Umgebung bezogen, sie können daher global genutzt werden.

Um aus dem MiaW Aktionen im Hauptfilm auszulösen oder Properties auszulesen, haben wir folgende Möglichkeiten:

- System-Properties, globale Variablen  
System-Properties (wie `the colordepth`, `the platform` etc.) und globale Variablen stehen dem MiaW ebenso zur Verfügung wie der Bühne.
- Besetzungen und Darsteller  
Die Analyse und Manipulation von Besetzungen und Darstellern müssen in ein `tell the stage`-Konstrukt eingebunden werden:

```
tell the stage
  put the number of castlibs
  put the activeCastlib
  put the number of members of castlib \
    (the activeCastlib)
  put member(1,1).name -- etc.
end tell
```

- Score und Sprites

Alle frame-bezogenen Abfragen und Befehle müssen ebenfalls in ein `tell the stage` eingebunden werden:

```
tell the stage
  put the frame
  go to frame 1
  put the frameLabel
  put marker(0) -- etc.
end tell
```

Dasselbe gilt für alle sprite-bezogenen Lingo-Statements:

```
tell the stage
  put sprite(4).type
  put sprite(4).member.name
  put sprite(4).endFrame - sprite(4).startFrame
  put sprite(4).scoreColor
  put the lastchannel -- etc.
end tell
```

the score (als Referenz auf die Drehbuchinformationen) ist ebenfalls objektspezifisch:

```
tell the stage to set temp = the score
filmloop1 = new(#filmloop)
filmloop1.media = temp
```

Der Skriptausschnitt kopiert den Score des Hauptfilms in einen Darsteller des MiaWs.

#### ■ Auswahlen

the scoreSelection und the selection of castlib - als Properties, die nur beim Authoring relevant sind - können dagegen ohne umschließendes tell the stage genutzt werden.

```
put the scoreselection
-- [[5, 5, 1, 2], [6, 6, 1, 2], [9, 9, 1, 2]]
```

zeigt die Auswahl im Drehbuch des Hauptfilms an, auch wenn die Property im MiaW abgefragt wird. Hier sind die Sprites 5, 6 und 9 jeweils von Bild 1 bis 2 ausgewählt;

```
put the selection of castlib 1
-- [[1, 3], [7, 7], [9, 10]]
```

beschreibt eine Auswahl im Fenster der ersten Besetzung, welche die Darsteller 1, 2, 3, 7, 9 und 10 umfasst.

#### „Frameworks“ für MiaW-Tools und Skripte

Bei vielen MiaW-Tools – und auch bei einfachen Skripten, die beim Authoring erstellt werden, um bestimmte Arbeiten zu vereinfachen – ist es häufig nötig, Befehle auf eine bestimmte Auswahl an Darstellern anzuwenden. Beispiele sind z.B. die Skripte zur Überprüfung der Darstellerfarbpaletten, zum Setzen der Regpoints von Bitmap-Darstellern oder zum Batch-Formatieren von Textdarstellern.

**Auswahl von Darstellern einer Besetzung** • Das folgende Gerüst dient dazu, auf die Auswahl von Darstellern in der aktiven Besetzung einen (oder mehrere) Befehle anzuwenden. Durch das tell the stage-Konstrukt werden bei Verwendung im MiaW nicht die Darsteller des MiaW verändert, sondern die des Hauptfilms:

```
on castsel_do param1 -- {, param2, param3}
  tell the stage
    temp = castlib(the activeCastlib).selection
    repeat with i = 1 to temp.count
      repeat with memnum = temp[i][1] to temp[i][2]
        memref = member(memnum, the activeCastlib)
        --hierher einen Handler-Aufruf, z.B.:
        doStuff memref, param1 --{, param2, param3}
      end repeat
    end repeat
  end tell
end
```

*Listing 9.7: Grundgerüst für MiaW-Tools (aktuelle Darstellerauswahl; Filmskript)*

Das Skript durchläuft alle Sublisten der selection of castlib. Entscheidend ist, dass es in der inneren repeat-Schleife alle Darstellerreferenzen der betreffenden Besetzungsauswahl zur Verfügung stellt (Variable memref); in dem Handler doStuff können diverse Befehle diese Information nutzen, z.B.:

#### ■ Regpoint setzen:

```
-- Aufruf: castsel_do point(100,100)

on doStuff memref, param1
  memref.regPoint = param1
  put string(memref) && "ok"
end
```

*Listing 9.8: Regpoint setzen  
(für MiaW-Tools, Filmskript)*

#### ■ Farbtiefe und -palette überprüfen:

```
-- Aufruf: castsel_do 8, #systemWin

on doStuff memref, param1, param2
  if memref.depth > param1 then put \
    "Farbtiefe >"&&param1&&":"&&string(memref)
  if memref.paletteref <> param2 then put \
    "Palette"&&paletteref&":"&&string(memref)
end
```

*Listing 9.9: Farbtiefe und-palette überprüfen  
(für MiaW-Tools, Filmskript)*

#### ■ Externe, gelinkte Medien importieren:

```
-- Aufruf: castsel_do [#bitmap, #flash]

on doStuff memref, param1
  if getpos(param1, memref.type) > 0 then
    -- Typ entspricht einem der in param1 genannten
    thefile = memref.filename
    if thefile <> EMPTY then
      -- Datei ist extern
      importFileInto memref, thefile
      put string(memref)&&"importiert aus"&&thefile
    end if
  end if
end
```

*Listing 9.10: Gelinkte Medien importieren  
(für MiaW-Tools, Filmskript)*

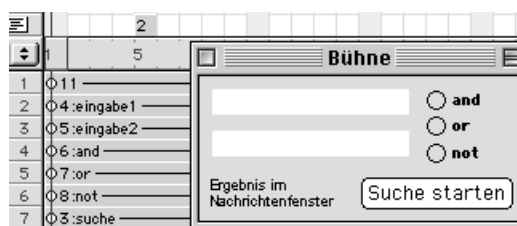
#### ■ Boolesche Textsuche

Bei dieser Suche sollen zwei Suchstrings durch logische Operatoren verknüpft werden können. Groß- und Kleinschreibung spielen bei String-Vergleichen in Lingo keine Rolle. Die Darstellerreferenz wird hier genutzt, um den type abzufragen und den Textinhalt (memref.text) mit den Such-Strings zu vergleichen:

```
-- Aufruf: castsel_do "hans", #not, "peter"

on doStuff memref, param1, param2, param3
  if memref.type = #text then
    flag1 = 0
    flag2 = 0
    ergebnis = 0
    if memref.text contains param1 then flag1 = 1
    if memref.text contains param3 then flag2 = 1
    case param2 of
      #not: ergebnis = (flag1 AND NOT(flag2))
      #or: ergebnis = (flag1 OR flag2)
      #and: ergebnis = (flag1 AND flag2)
    otherwise
      beep
    end case
    if ergebnis = 1 then put string(memref)
  end if
end
```

*Listing 9.11: Boolesche Textsuche  
(für MiaW-Tool, Filmskript)*



*Abbildung 9.13: Aufbau der Bühne für das Beispiel „Boolesche Suche“*



Abbildung 9.14: Das Behavior „Auswahlfeldgruppe“ aus der Bibliothek

Wie wird aus solchen Handlern ein MiaW-Tool? Das letzte Beispiel scheint uns für die praktische Umsetzung am verlockendsten:

- Auf einer kleinen Bühne (z.B. 200 x 90) platzieren wir zwei editierbare Textdarsteller („Rahmen: Fest“, „Bearbeitbar“, „Mit Tabulatortaste zum nächsten Element“), drei Radio-Buttons und einen Button für `SUCHE STARTEN` (Abbildung 9.13.)

- Die drei Radio-Buttons erhalten alle das Skript „Auswahlfeldgruppe“ aus der BIBLIOTHEK (Abteilung „Controls“). Das „and“-Behavior wird wie folgt konfiguriert (Abbildung 9.14):

- Die anderen beiden Buttons bekommen denselben Eintrag unter „Kennung“, um sie als Gruppe zusammenzufassen, sollen aber ausgeschaltet, also „Aus“ sein.

- Der Button `SUCHE STARTEN` erhält das folgende Skript:

```
on mouseup me
  radbut = sendAllSprites \
    (#RadioGroup_SelectedButton, \
    "Auswahlfeldgruppe3")
  param1 = member("eingabe1").text
  param2 = symbol(radbut.name)
  param3 = member("eingabe2").text
  castsel_do param1, param2, param3
end
```

Listing 9.12: Aufrufbutton für Boolesche Suche (Behavior-Skript)

Er ermittelt also die Parameter für den Funktionsaufruf `castsel_do` (siehe oben) und ruft die Funktion dann auf. Der Handler `#RadioGroup_SelectedButton` ist im Behavior „Auswahlfeldgruppe“ bereits angelegt.

- Im Filmskript legen wir folgende Bestandteile ab: das Skript `castsel_do` (siehe weiter oben in diesem Abschnitt) mit drei Übergabeparametern

```
on castsel_do param1, param2, param3
```

sowie das `doStuff`-Skript für die Boolesche Textsuche. Die Ergebnisse sollen weiterhin in das Nachrichtenfenster ausgegeben werden.

- Außerdem sind noch folgende neue Teilskripte in ein Filmskript zu schreiben:

```
on preparemovie
  (the activeWindow).windowtype = 49
  (the activeWindow).title = "Suche..."
  repeat with i in [4,5]
    member(i,1).text = " "
  end repeat
end

on closeWindow
  forget(the activeWindow)
end
```

Listing 9.13: MiaW-Setup (Filmskript)

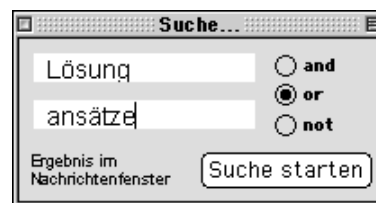


Abbildung 9.15: Das Ergebnis: das MiaW-Tool Boolesche Textsuche

Im `prepareMovie`-Handler wird die Optik des MiaWs verbessert (`windowType`, `title`) und sichergestellt, dass die Eingabefelder leer sind. Im `close-`

Window-Handler räumen wir den Speicher auf, indem wir das Fenster wirklich löschen (forget Fensterreferenz). Näheres zum Aufräumen erfahren Sie im nächsten Abschnitt.



Das komplette Beispiel finden Sie im Verzeichnis WORKSHOP/KAPITEL09/SUCHE auf der CD-ROM

Wenn Sie den Director-Film kompakt sichern und in den XTRAS-Ordner der Director-Applikation kopieren, steht Ihnen das Tool beim nächsten Programmstart zur Verfügung. Achten Sie darauf, dass Sie einige Textdarsteller im Besetzungsfenster ausgewählt haben, wenn Sie die Suche ausprobieren.

Das Beispiel auf der CD-ROM fängt noch einige Fehler ab (kein Eintrag im zweiten Feld, kein Textdarsteller ausgewählt) und gibt etwas mehr Rückmeldung ins Nachrichtenfenster aus (vgl. Abschnitt AUFRÄUMEN).

**Alle Darsteller einer oder aller Castlibs** • Ersetzen Sie den Handler `castsel_do` aus dem vorangegangenen Abschnitt durch `castall_do`, wenn Sie alle Darsteller einer Castlib als Darstellerreferenz für einen Handler-Aufruf erhalten wollen:

```
on castall_do param1 -- {, param2, param3}
  tell the stage
    num = the number of members of castlib \
      (the activeCastlib)
    repeat with memnum = 1 to num
      memref = member(memnum, the activeCastlib)
      --hierher einen Handler-Aufruf, z.B.:
      doStuff memref, param1 -- {, param2, param3}
    end repeat
  end tell
end
```

Listing 9.14: Grundgerüst für MiaW-Tools (aktuelle Besetzung; Filmskript)

Filtern Sie dann im `doStuff`-Handler zum Beispiel nach Darstellertyp oder anderen Spezifika, zum Beispiel so:

```
if (memref.type = #text) then ...
```

Sollen alle Darsteller aller vorhandenen Besetzungen durchforstet werden, so ist eine weitere repeat-Schleife einzufügen:

```
on allcasts_allmembers_do param1 -- {, param2, \
  param3}
  tell the stage
    numlibs = the number of castlibs
    repeat with libnum = 1 to numlibs
      num = the number of members of castlib libnum
      repeat with memnum = 1 to num
        memref = member(memnum, libnum)
        --hierher einen Handler-Aufruf, z.B.:
        doStuff memref, param1 -- {, param2, param3}
      end repeat
    end repeat
  end tell
end
```

Listing 9.15: Grundgerüst für MiaW-Tools (alle Besetzungen; Filmskript)

**Drehbuch- und Sprite-Referenzen** • Das folgende Skript liefert das Grundgerüst, um regelmäßig zu überprüfen, was das gerade im Drehbuch ausgewählte Sprite und der diesem Sprite zugeordnete Darsteller ist. Es kann wiederum für verschiedene Zwecke genutzt werden:

```
property spriteref, memref, display
on prepareFrame me
  tell the stage
    if not (the scoreSelection = []) then
      newspriteref = \
        sprite((the scoreSelection)[1][1])
      newmemref = newspriteref.member
      -- falls unverändert, dann Abbruch
      if (newspriteref = spriteref) AND
        (newmemref = memref) then return
      spriteref = newspriteref
      memref = newmemref
    end if
  end tell
end
```

```

-- Hier einen Handler einklinken, z.B.:
myShowProps me
end if
end tell
member("display").text = display
end

```

Listing 9.16: *Sprite-Referenzen auslesen (Frame-Skript)*

Der Handler `myShowProps` nutzt die als Property gesicherten `Sprite`- und `Darsteller`referenzen und zeigt einige Eigenschaften für beide an:

```

on myshowprops me
display = \
string(spriteref)&&string(memref)&return& \
"-----" & \
return
display = display & "• Sprite-Properties:" & \
return
display = display & \
"color:"&&string(spriteref.color) & return
display = display & \
"bgcolor:"&&string(spriteref.bgcolor) & return
display = display & \
"loc:"&&string(spriteref.loc) & return
display = display & \
"rect:"&&string(spriteref.rect) & return
displaytext = displaytext \
& "-----" & return
display = display & \
"• Darsteller-Properties:" & return
display = display & \
"regpoint:"&&string(memref.regpoint) & return
display = display & \
"rect:"&&string(memref.rect) & return
end

```

Listing 9.17: *Darsteller- und Sprite-Properties auslesen (für MiaW-Tool, Frameskript)*

Das Ergebnis wäre ein kleines Analyse-MiaW wie in Abbildung 9.16. Anstatt die Property-Namen in das Skript zu schreiben, können Sie natürlich auch einen Konfigurations-Screen mit Auswahl- oder Eingabemöglichkeit für die Properties erstellen, die Sie interessieren. Mit

```
value("spriteref."&einPropString)
```

können Sie Properties abfragen, deren Namen Ihnen nur als String vorliegen.

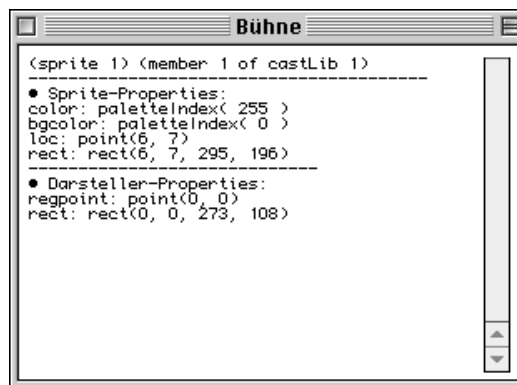


Abbildung 9.16: *Analyse von Sprites*

Soll nicht die Auswahl im Drehbuch, sondern das gerade unter der Maus befindliche Sprite angezeigt werden, so müssen Sie den `prepareFrame`-Handler folgendermaßen ändern:

```

property spriteref, memref, display
on prepareFrame me
tell the stage
if (rollover() = 0) then
displaytext = ""
spriteref = 0
memref = 0
else
newspriteref = sprite(rollover())
newmemref = newspriteref.member
-- falls unverändert, dann Abbruch
if (newspriteref = spriteref) AND
(newmemref = memref) then return
spriteref = newspriteref
memref = newmemref
-- Hier einen Handler einklinken, z.B.:
myShowProps me
end if
end tell
member("display").text = display
end

```

Listing 9.18: *Sprites unter der Maus analysieren (Frame-Skript)*

Dieses Skript funktioniert allerdings nur, wenn der Hauptfilm abgespielt wird.

### Aufräumen

Wenn Sie MiaW-Tools nur für sich erstellen, kann Ihnen das Thema „Aufräumen“ relativ egal sein. Wenn Sie dagegen nicht absehen können, wer alles Ihr Tool benutzt, sollten Sie dafür sorgen, dass es möglichst wenig „Rückstände“ hinterlässt. Achten Sie besonders auf folgende Punkte:

- Löschen Sie – am Besten im Event-Handler `on closeWindow` – die Fensterreferenz. Die Lingo-Zeile

```
set the windowList = []
```

löscht alle Fenster; sie ist hier nicht geeignet, da Sie nicht wissen, ob der Anwender noch andere Fenster geöffnet hat. Sinnvoller ist `forget(the activeWindow)` oder `deleteAt(the windowList, the activeWindow)`.

- Achten Sie darauf, dass Sie nicht mit globalen Variablen um sich werfen. Diese bleiben erhalten, auch wenn das Fenster längst geschlossen ist. Eine Löschroutine, in der Sie alle benutzten globalen Variablen gleich `VOID` setzen, ist auf jeden Fall sinnvoll. Im Beispiel „Boolesche Textsuche“ sieht das wie folgt aus (`textflag` und `foundflag` werden im Handler `doStuff` gesetzt, wenn mindestens ein Textdarsteller ausgewählt ist bzw. wenn mindestens eine Fundstelle auftritt):

```
global textflag, foundflag

on castsel_do param1, param2, param3
  -- (...)
  if textflag <> 1 then \
    put "Kein Textdarsteller ausgewählt."
  else if foundflag <> 1 then \
    put "Kein passender Darsteller gefunden."
  textflag = VOID
  foundflag = VOID
end
```

*Listing 9.19: Globale Variablen auf VOID setzen*

Sobald eine globale Variable auf `VOID` gesetzt wurde, wird sie mit dem Befehl `showGlobals` nicht mehr angezeigt. In der System-Property `the globals` ist ihr Eintrag aber weiterhin vorhanden. Senden Sie nicht den Befehl `clearGlobals` – der Anwender hat sicher auch eigene globale Variablen, die er nicht verlieren möchte.

- Wenn Sie objektorientiert programmieren, bedenken Sie, dass ein Objekt erst gelöscht wird, wenn keine Referenzen mehr auf dieses Objekt bestehen. Insbesondere in der `actorList` verbleibt oft unfreiwillig eine Referenz, die Sie mit `(the actorList).deleteAt(IhreReferenz)` beseitigen können.

## Beispiele für MiaW-Tools

### Behavior-Helper

Jeder Programmierer stellt irgendwann mehr oder weniger entnervt fest, dass er die gleichen Prozedurdefinitionen und Befehlsfolgen wieder und wieder in das Skriptfenster von Director eintippt, und sucht nach Abhilfe. Eine Möglichkeit ist die Installation eigenständiger Tools wie z.B. „TypeIt4me“ für den Mac oder „Quickkey“ für Windows. Programme dieser Art gestatten es, oft benötigte Textpassagen per Hotkey oder Mausklick in das aktuelle Skript einzufügen. Ihr Einsatzgebiet ist selbstverständlich nicht auf Director beschränkt.

Für MacOS gibt es außerdem das `Autocomplete-Xtra`, das Lingo-Keywords vervollständigt, aber auch ganze Prozedurdefinitionen in das Skriptfenster einfügen kann (vgl. Kapitel 10 XTRAS → 589).

Eine andere Möglichkeit besteht in der Anlage einer externen Besetzungsdatei, in der man die regelmäßig benötigten Prozeduren oder Prozedurteile speichert. So eine Toolbox wird nicht mit den Filmen eines bestimmten Projekts verknüpft, da in der Regel notwendige Anpassungen der Skripttexte die universelle Einsetzbarkeit beeinträchtigen würden. Vielmehr werden die Skriptdarsteller dieser Besetzungsdatei bei Be-

darf kopiert und in eine Besetzung des aktuellen Filmes eingefügt.

Als Movie-in-a-Window laufende Tool-Xtras wie der Behavior-Writer von Roy Pardi (<http://www.tiac.net/users/rpardi/behaviorwriter/>, Abbildung 9.10) → 521 oder wie unser auf der CD enthaltener Behavior-Helper nutzen die Möglichkeit, Behaviors aus Lingo heraus zu erzeugen.



Das Arbeitsbeispiel finden Sie auf der CD-ROM im Verzeichnis WORKSHOP/KAPITEL09/BEHAVIORHELP

Während sich die Funktion des Behavior-Helpers auf die Eingabe des sich prinzipiell wiederholenden Behavior-Grundgerüsts beschränkt, erlaubt der Behavior-Writer auch das komfortable Definieren und Setzen von Property-Variablen. Beide Tools funktionieren aber nach demselben Prinzip. Sie generieren Behaviors, indem sie einen neuen Darsteller vom Typ #script erzeugen, den Skripttyp auf #score (also „Behavior“) setzen und die Kopie eines intern gespeicherten Textfelds als Skripttext einsetzen.

```
newmem = new(#script)
newmem.scriptType = #score
newtxt = member( "copyscripttext" ).text
newmem.scriptText = newtxt
activateScriptEditor newmem.membNum, \
    newmem.castLibNum, 0, 0
```

Um das neu erzeugte Behavior automatisch zur Bearbeitung im Skripteditor zu öffnen, wird die Funktion activateScriptEditor des im Lieferumfang enthaltenen Xtras UIHelper benötigt.

### Anzeige der Soundkanäle



Die Beispieldatei finden Sie auf der CD-ROM im Verzeichnis WORKSHOP/KAPITEL09/SIMPLESOUND

Die Aufgabe des Miaw-Tools SimpleSound ist die Anzeige der Belegung der Soundkanäle während des Authorings und im Projektor. Insbesondere in umfangreichen Projekten, in denen eine Vielzahl von Musikstücken, Stimmen, Soundeffekten usw. miteinander gemischt wird, ist es praktisch, auf einen Blick sehen zu können, welcher Soundkanal gerade was mit welcher Lautstärke wiedergibt.

| - simpleSound - |      |        |               |
|-----------------|------|--------|---------------|
|                 | busy | volume | member        |
| snd 1           | 0    | 100    |               |
| snd 2           | 1    | 150    | Song05a       |
| snd 3           | 1    | 255    | Song06a       |
| snd 4           | 0    | 255    |               |
| snd 5           | 0    | 255    |               |
| snd 6           | 0    | 255    |               |
| snd 7           | 0    | 255    |               |
| snd 8           | 1    | 255    | externe Datei |

Abbildung 9.17: Das Authoringtool SimpleSound

Für die Anzeige der Sounddarsteller ist eine Besonderheit zu beachten: Sie können die drehbucheigenen Soundkanäle, also Sound 1 und Sound 2, mit the frameSound1 und the frameSound2 erhalten, ohne dass der Sound spielen muss oder der Abspielkopf in Bewegung ist. Die weiteren Soundkanäle haben keine solche Referenz im Drehbuch; allerdings können sie eine Darstellerreferenz zurückliefern, falls in ihnen ein Sound spielt, der ein Darsteller ist. soundbusy(Kanalnummer) kann TRUE zurückliefern, ohne dass der Sound mit einem Darsteller verbunden ist. Dann wird gerade eine externe Datei abgespielt.

Der Film verwendet die drei globalen Variablen gsndnames, gsndstates und gsndvolumes und den Befehl tell the stage zur Abwicklung der Kommunikation mit dem Hauptprogramm:

```

on startMovie
  (the activewindow).windowtype = 49
end

on getsoundinfo
  gsndnames = ""
  gsndstates = ""
  gsndvolumes = ""
  tell the stage

    -- Soundbelegung in Drehbuch-Kanälen überprüfen
    if the frameSound1 <> 0 then put \
      member(the frameSound1).name into \
      gsndnames.line[1]
    if the frameSound2 <> 0 then put \
      member(the frameSound2).name into \
      gsndnames.line[2]
    repeat with snd = 1 to 8
      -- Darsteller über puppetsound/playlist?
      if objectP(sound(snd).member) then
        put string(sound(snd).member.name) into \
        gsndnames.line[i]
      else
        -- kein Darsteller; sound playfile
        if soundbusy(i) then put "externe Datei" \
        into gsndnames.line[i]
      end if
      put string(soundBusy(snd)) into \
      gsndstates.line[i]
      put string(sound(snd).volume) into \
      gsndvolumes.line[i]
    end repeat
  end tell
  member("sndstates").text = gsndstates
  member("sndvolumes").text = gsndvolumes
  member("sndnames").text = gsndnames
end

```

*Listing 9.20: Belegung der Soundkanäle analysieren (für MiaW-Tools, Filmskript)*

Da getSoundinfo von der Prozedur enterFrame des Filmskripts aufgerufen wird, bestimmen Sie durch die im Tempokanal gewählte Bildrate, wie oft ein Update der Anzeige erfolgt.

```

on enterFrame
  getSoundinfo
end

```

### RAMLight

Immer langsamer werdende Animationen, stockender oder fehlender Sound... Was macht der Projektor mit dem RAM? Ein Tool, das diese Frage beantworten hilft, indem es den freien RAM im Authoring- und Projektormodus anzeigt, stellt die Firma Dreamlight unter <http://www.dreamlight.com/webshop/software/ramlight.htm> mit der Shareware RAMLight zum Preis von ca. 10 US-Dollar zur Verfügung.



Abbildung 9.18: Die Shareware „RAMLight“ von Dreamlight

RAMLight ist ein Shockwave-Movie, das vom zu untersuchenden Hauptprogramm als Movie-in-a-Window (MiaW) geöffnet werden muss. Als zusätzliche Features bietet RAMLight Load- und Unload-Funktionen für Frames und Darsteller.

Wenn man die letzteren Funktionen einfacher gestaltet, auf die grafische Darstellung des freien Speichers verzichtet und sich mit einer simplen Textanzeige begnügt, ist die Realisierung eines ähnlichen Tools eine einfache Angelegenheit.

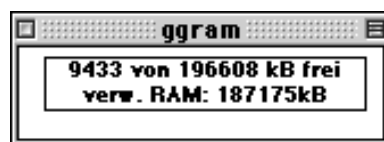


Abbildung 9.19: Die Eigenbau-Alternative besitzt den Vorteil der Erweiterbarkeit.

Erzeugen Sie zuerst einen neuen Film mit möglichst kleiner Bühnengröße (ca. 160 x 40 Pixel), platzieren Sie ein z.B. "display" benanntes Textfeld, und schreiben Sie ein Skript, das regelmäßig den freien Speicher in diesem Feld anzeigt.

```
on exitFrame
  put string(the freeBlock/1024) &" von " & \
    string(the memorySize/1024) &" kB frei" \
    & RETURN &"verw. RAM: " & \
    string(the memorySize / 1024 - the freeBlock\
    / 1024) &"kB" into field "display"
end
```

*Listing 9.21: Freien Speicher anzeigen  
(Film- oder Frame-Skript)*

Wie häufig dieses Frame-Skript die Anzeige aktualisiert, lässt sich über die Bildrate im Tempokanal des Drehbuchs steuern.

Um mit einem Klick auf die Speicheranzeige alle im Hauptprogramm entbehrlichen Daten aus dem Arbeitsspeicher löschen zu können, benötigt das Textfeld "display" ein Behavior mit folgendem Inhalt:

```
on mouseUp
  tell the stage
    unload
  end tell
end
```

Im Projektor lässt sich das so erzeugte Tool mit dem an beliebiger Stelle zu platzierenden Befehl `open window MiaWDateiname` öffnen. Soll sich das Tool mit einem Datei-Öffnen-Dialog laden lassen, genügt die Angabe `open window ""`. Beide Befehle können auch in der LINGO.INI (vgl. unten → 539) stehen.

## Window Type Explorer

Zu Fenstertypen vgl. auch Kapitel 8, PLATTFORM SPEZIFISCHE ARBEITSTECHNIKEN → 470

Der Fenstertyp eines im eigenen Fenster laufenden Filmes oder Movies-in-a-Window (MiaW) wird von der Eigenschaft `window(w).windowType` bestimmt. Diese Eigenschaft beschreibt den Typ des Fensters als Dezimalzahl. Wenn Sie sich beim regelmäßigen Nachschlagen dieser Dezimalcodes in der Online-Hilfe bzw. im Handbuch und dem Ausprobieren der verschiedenen Fenstertypen erappen, können Sie sich schnell und einfach eine komfortable Gedächtnisstütze in Form eines MiaW-Xtras bauen, das sich bei Bedarf über einen Eintrag im Menü XTRAS aufrufen lässt.



Abbildung 9.20: Der Window Type Explorer erleichtert die Auswahl des Fenstertyps.

Alles, was Sie benötigen, ist ein Film mit kleinerer Bühnengröße (z.B. 432 x 220 Pixel), der die verschiedenen Fenstertypen und ihre Dezimalcodes in einem scrollbaren Textfeld enthält. Diese Textdaten lassen sich übrigens einfach aus der Online-Hilfe kopieren. In diesem Beispiel wurden lediglich die in der eingefügten Kopie enthaltenen Leerräume durch einen Doppelpunkt als Trennzeichen zwischen Fenstercode und -beschreibung ersetzt. Das folgende Behavior ist dem "windowtypes" benannten Textfeld zugeordnet und sorgt dafür, dass das MiaW den per Mausklick ausgewählten Fenstertyp annimmt.

```

on mouseDown me
  ml = the mouseLine
  if ml < 1 then exit
  id = the itemDelimiter
  the itemDelimiter = ":"
  hilite line ml of field "windowtypes"
  wtype = value(item 1 of line ml of field \
    "windowtypes")
  (the activeWindow).windowType = wtype
  put wtype into field "wtype"
  hilite line ml of field "windowtypes"
  the itemDelimiter = id
end

```

*Listing 9.22: Windowtype ändern (Behavior-Skript)*

Um das MiaW-Tool im Menü XTRAS zu installieren, kopieren Sie auch hier einfach die entsprechende Filmdatei in den Ordner XTRAS im Director-Ordner und starten Director neu.

## Makros

Ich persönlich ärgere mich immer über Dinge, die ich mehrfach tippen muss. Da schreibe ich lieber einmal

```

on openWindow
  the windowList = []
  put return & "put the windowlist" & return & \
    "--" && the windowList
end

```

*Listing 9.23: Windowlist löschen  
(für Makro-Tool, Filmskript)*

anstatt fünfmal am Tag set the windowlist = [] in das Nachrichtenfenster einzugeben. Ein anderes schönes Beispiel ist folgendes:

```

on openWindow
  y = new(xtra "fileIO")
  filePath = y.displayOpen()
  put filePath
  forget(the activeWindow)
end

```

*Listing 9.24: Dateipfade ausgeben  
(für Makro-Tool, Filmskript)*

Es öffnet mit Hilfe des Xtras FileIO einen Datei-Öffnen-Dialog und gibt den Pfad der ausgewählten Datei in das Nachrichtenfenster aus, was sehr praktisch ist, um die exakten Pfade von externen Dateien zu erhalten.

Das openWindow weist bereits darauf hin: Makros sind MiaWs, die kein User-Interface benötigen und sich sofort nach dem Ausführen der angegebenen Befehle beenden können. Schreiben Sie eines der Skripte als Filmskript in einen Director-Film, machen Sie die Bühne möglichst klein, und sichern Sie den so erstellten Director-Film in das XTRAS-Verzeichnis.

Ein solches Makro ist grundsätzlich auch geeignet, um die Instanz eines Parent-Skripts im Speicher zu erzeugen (vgl. Kapitel 2 SKRIPTING-GRUNDLAGEN → 148).

## Text extrahieren



Auf CD-ROM:  
WORKSHOP/KAPITEL09/MIAW\_EXTRACT

Das folgende Beispiel nutzt eine Skriptinstanz. Es ist im Prinzip wie die MiaW-Beispiele weiter oben im Kapitel aufgebaut. Wenn Sie im prepareMovie-Handler die Zeile

```
doit = VOID
```

entfernen, wird die Instanz beim Schließen des Fensters nicht gelöscht, so dass Sie zukünftig mit dem Befehl `doit.extract()` eine global verfügbare neue Lingo-Funktion zur Hand haben, mit der Sie Skript- und Darstellertexte ins Nachrichtenfenster ausgeben lassen können. Wenn Sie es (wie ich) vorziehen, dafür lieber das Makro erneut aufzurufen, sollten Sie `doit = VOID` setzen und damit die Skriptinstanz nach einmaligem Ausführen löschen.

```

global doit
on preparemovie
  mywindow = the activeWindow
  doit = new(script "myparent")
  tell the stage to doit.extract()
  doit = VOID -- vgl. Text
  forget mywindow
end

```

Listing 9.25: Objektinstanz erzeugen (Filmskript)

Das folgende Parent-Skript heißt "myparent"; es extrahiert den (Skript-)Text aller ausgewählten Darsteller, die dem Typ des ersten ausgewählten Darstellers entsprechen.

```

property texthalter
on new me
  return me
end

on extract me
  temp = castlib(the activeCastlib).selection
  mytype = member (temp[1][1], (the \
    activeCastlib)).type
  if getpos([#text,#field,#script], mytype) = 0 \
  then return
  texthalter = ""
  --
  repeat with i = 1 to temp.count
    repeat with memnum = temp[i][1] to temp[i][2]
      memref = member(memnum, the activeCastlib)
      doStuff me, memref, mytype
    end repeat
  end repeat
  put texthalter
  texthalter = ""
end

on doStuff me, memref, param1
  if memref.type = param1 then
    if param1 = #script then
      texthalter = texthalter & return & return & \
        memref.scriptType && string(memref) && \
        memref.name & return & \
        "-----" & \
        return & return & memref.scriptText

```

```

else
  texthalter = texthalter & return & return & \
    string(memref) && memref.name &return & \
    "-----" & \
    return & return & memref.text
end if
end if
end

```

Listing 9.26: Text extrahieren (Parent-Skript)

### Score-To-List



Auf CD-ROM:  
WORKSHOP/KAPITEL09/MIAW\_GETPATHOFSPRITE

Manchmal ist es ausgesprochen hilfreich, die im Drehbuch ausgelegte Animation eines Sprites (mit Tweening etc.) in eine Form zu überführen, die es erlaubt, die gesamte Animation aus Lingo heraus ablaufen zu lassen. Das folgende Skript ist ein Makro, das die Position des ausgewählten Sprites in eine Liste schreibt und diese dann im Nachrichtenfenster ausgibt. Selbstverständlich können Sie auch andere Sprite-Eigenschaften auf diese Weise auslesen.

```

on startmovie
  tell the stage to doStageStuff
end

on stopmovie
  deleteone(the windowlist, the activewindow)
end

on dostagestuff
  if the scoreSelection = [[]] then
    alert "Bitte ein Sprite auswählen."
    return
  end if
  selectedsprite = sprite(the scoreselection[1][1])
  -- erstes ausgewähltes Sprite
  fromframe = the scoreselection[1][3]
  toFrame = the scoreselection[1][4]
  theList = []

```

```
--
repeat with i = fromframe to toFrame
  go to frame i
  thelist.append(selectedsprite.loc)
end repeat
put thelist -- Ausgabe ins Nachrichtenfenster
end
```

*Listing 9.27: Score-To-List  
(für MiaW-Tool, Filmskript)*

### getBehaviorToolTip als Makro-Interface




Auf der CD-ROM finden Sie eine Beispielbesetzung im Verzeichnis WORKSHOP/KAPITEL09/MAKROS

Die folgende Technik zweckentfremdet den Handler `getBehaviorToolTip`, der zum Standardinventar eines Behaviors gehört (vgl. Kapitel 2 → 142). Bei Behaviors in der Library (BIBLIOTHEK) wird er aufgerufen, um einen Erklärungstext anzuzeigen, wenn die Maus sich über das Behavior-Icon bewegt.

Da somit ein ganz normaler Lingo-Handler ausgeführt wird, können wir dort auch andere Dinge tun, als nur mit `return` einen Text zurückzuliefern. Wer möchte, kann im Verzeichnis LIBS im Director-Ordner eine eigene Castlib „Makros“ anlegen und dort Skripte wie die folgenden unterbringen:

```
on getBehaviorToolTip
  global xyz_xtralist
  if voidp(xyz_xtralist) OR the commanddown then
    xyz_xtralist = the xtralist
    put xyz_xtralist
  end if
  return "Puts the Xtralist. Rollover behavior. \
  Hold down command key to refresh."
end
```

*Listing 9.28: GetbehaviorToolTip-Makro 1  
(Behavior-Skript)*

Dieses Skript würde beim ersten Rollover bzw. wenn die /Ctrl-Taste gedrückt ist, die Xtra-

List in das Nachrichtenfenster schreiben und ansonsten einen Erklärungstext anzeigen.

Das folgende Skript demonstriert, dass mit diesem Verfahren auch andere Funktionen ausgeführt werden können:

```
on getBehaviorToolTip
  if the commanddown then
    put the scoreselection
    put the activecastlib
    newmem = new(#script)
    put newmem
  end if
  return "Hold down command key!"
end
```

*Listing 9.29: GetbehaviorToolTip-Makro 2  
(Behavior-Skript)*

Die Referenz ist stets der aktuelle Director-Film und nicht die Besetzung, in der sich das Skript befindet. Der Darsteller wird demnach in der aktiven Besetzung des Director-Filmes angelegt.

### UIHelper-Xtra



„not supported“ – wenn Macromedia ein Xtra so kennzeichnet, sollten Sie von einer Verwendung in Projektoren oder in Tools, die Sie vertreiben wollen, absehen.

Das UIHelper-Xtra ist für eigene Tools überaus nützlich. Es wird von Director selbst genutzt, um beispielsweise Fenster zu öffnen. Leider ist es „not supported“ und undokumentiert, und Macromedia rät von seiner Verwendung ab.

Für Tools, die Sie selbst nutzen, können Sie es dennoch hervorragend einsetzen. Den vollen Funktionsumfang erhalten Sie mit folgender Skriptzeile im Nachrichtenfenster angezeigt:

```
put interface (xtra "UIHelper")
```

Hier zählen wir nur einige Funktionen auf, die für MiaW-Tools besonders interessant sind:

■ `*activateWindow int windowId:` bringt das durch `windowID` bezeichnete Fenster nach vorn. Das ist

beispielsweise nützlich, um das Besetzungsfenster (3) oder das Drehbuch (5) nach vorn zu bringen.

■ `*activateMediaEditor` `int memberNum`, `int castLibNum`: öffnet den Editor für den jeweiligen Darsteller, also beispielsweise das Malfenster für Bitmaps.

■ `*activateScriptEditor` `int memberNum`, `int castLibNum`, `int selectionStart`, `int selectionEnd`: öffnet den Skripteditor und markiert den angegebenen Bereich.

`activateScriptEditor` 12, 2, 135, 150

markiert demnach die Buchstaben 135 bis 150 des Skriptdarstellers 12 in Besetzung 2.

■ `*stringsEqual` `string a`, `string b`: vergleicht zwei Strings und ist – anders als der Vergleich mit `-- case-sensitiv`.

```
put "hans" = "Hans"
-- 1
put stringsEqual("hans", "Hans")
-- 0
put stringsEqual("Hans", "Hans")
-- 1
```

■ `*setFileWritable` `string aFile`: nützlich, wenn Sie unter Windows Dateien von CD-ROM kopieren, da diese dann vom Betriebssystem als schreibgeschützt markiert werden.

## 9.4 Projektoren

### Universellen Startprojektor erstellen

Mit einem universell einsetzbaren Projektor lässt sich einige Zeit sparen. Anstatt beim Authoring und Testen immer wieder neue Projektoren mit dem ersten Film des Projekts zu erstellen, ist es immer sinnvoll, in den Projektor nur einen Minimalfilm (*Stub*) zu integrieren, der wiederum den ersten Film des Projektes aufruft.

Um dies zu erreichen, haben Sie mehrere Möglichkeiten:

■ Nennen Sie den ersten Film Ihres Projekts immer gleich. Dies ist die einfachste Variante, denn im Stub-Film, der in den Projektor integriert wird, können Sie dann einfach direkt auf diesen Film verzweigen:

```
on exitFrame
  go to movie "intro"
end
```

■ Nutzen Sie die LINGO.INI, um einer globalen Variable den Namen Ihres Startfilms zuzuweisen. Im Stub-Movie überprüfen Sie, ob die Variable gesetzt ist, und springen mit `go to movie` zum entsprechenden Film. Falls die Variable nicht definiert wurde, wird bei der folgenden Lösung der Dialog „Wo ist Movie“ angezeigt.

```
on startUp
  global myStartMovie
  mystartMovie = "intro"
end
```

*Listing 9.30: Universalprojektor 1 (Lingo.ini)*

```
on exitFrame
  global myStartMovie
  if not(voidP(myStartMovie)) then go to \
  movie mystartmovie
  else go to movie ""
end
```

*Listing 9.31: Universalprojektor 1 (Film- oder Frame-Skript)*

■ Eine Lösung, die ohne INI-Datei auskommt, ist die folgende: Der Universalprojektor ermittelt zuerst seinen Namen und versucht dann, einen gleichnamigen Director-Film im aktuellen Ordner zu laden. Dabei können Sie die Systemeigenschaft `the applicationName` nutzen.

```
on exitFrame
  the itemdelimiter = "."
  mystartmovie = (the applicationName).item[1]
  go to movie mystartmovie
end
```

*Listing 9.32: Universalprojektor 2 (Film- oder Frame-Skript)*

Der Stub-Film sollte natürlich die gleiche Hintergrundfarbe und Bühnengröße haben wie Ihr Startfilm, damit der Übergang nicht auffällt. Häufig werden Sie mit zwei Projektoren pro Plattform (640x480, 800x600, Hintergrundfarbe Schwarz) sehr weit kommen.

Xtras sollten Sie grundsätzlich nicht in den Projektor integrieren (vgl. → den nächsten Abschnitt); mit einem externen Xtras-Ordner neben Ihrem Projektor erreichen Sie nicht nur schneller startende Projektoren, sondern können auch flexibler zusätzliche Xtras integrieren oder wieder entfernen.

Weitere Optionen von Projektoren können Sie mit der Datei PROJEKTORNAME.INI einstellen (vgl. DIE INI-DATEIEN → 540).

### Schnell startende Projektoren erstellen

Auf dem Mac sind keine besonderen Vorkehrungen nötig, um den Projektorstart zu beschleunigen. Wählen Sie in den Erstellungsoptionen PLAYER: STANDARD. Windows dagegen benötigt zum Starten eines Standardprojektors deutlich länger, da die Anwendung zunächst alle integrierten DLLs und Xtras auspacken und temporär zwischenspeichern muss.

Das folgende Verfahren ist daher für jeden Windows-Projektor zu empfehlen. Es ist zwingend, wenn Ihre Anwendung auf Windows NT laufen soll und Sie nicht absehen können, ob der Anwender Schreibbefugnis für das Windows-Verzeichnis hat.

Ein Schnellstart-Projektor wird über die Option PLAYER: SHOCKWAVE im Dialogfeld PROJEKTOR ERSTELLEN... erstellt. Außerdem müssen Sie sicherstellen, dass im Dialogfeld MODIFIZIEREN/FILM/XTRAS keine Einträge vorhanden sind, bzw. die Option „In Projektor einschließen“ ausgeschaltet ist. Ein solcher Projektor wird demnach keine Xtras enthalten, und die nötigen DLLs werden nicht in den Projektor integriert.



Damit er dennoch lauffähig ist, müssen die DLL-Dateien DIRAPI.DLL, IML32.DLL, MSVCRT.DLL und PROJ.DLL im Projektor-

Verzeichnis liegen; Sie müssen ausserdem einen XTRAS-Ordner anlegen und alle benötigten Xtras hineinkopieren. Abbildung 9.21 zeigt ein solches Programmverzeichnis. Die INI-Dateien sind nicht notwendig.

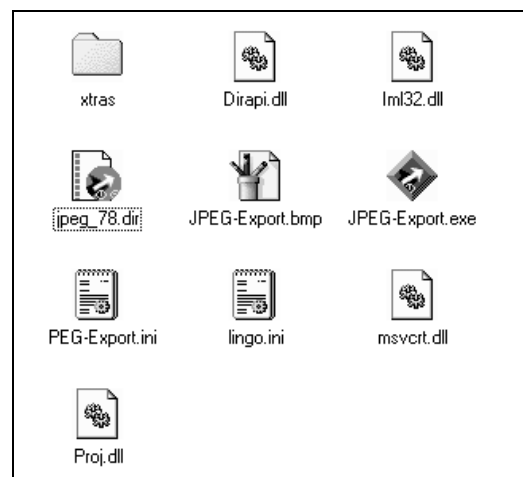


Abbildung 9.21: Setup für einen schnell starten den Windows-Projektor. Die DLL-Dateien dürfen in Director 8 auch in den Xtras-Ordner verfrachtet werden.

Wenn Sie es gerne aufgeräumter haben, können Sie in Director 8 die DLL-Dateien auch in den XTRAS-Ordner verschieben.

### Faststart-Projektoren in Director 7 und 8

In Director 7 gab es mehrere Probleme mit Faststart-Projektoren, die zu einer ganzen Serie von Technotes und Updates derselben und zu einiger Verwirrung geführt haben.

■ Die Meldung "This application requires the Shockwave Player which is not installed..." wurde für fast jeden Fehler beim Start eines Projektors ausgegeben und war insofern zumeist irreführend. Dies sollte in Director 8 nicht mehr auftreten.

■ Die Microsoft-Library MSVCRT.DLL war nötig, damit ein Projektor starten konnte. Macromedia hat erst nach Auslieferung von Director 7 be-

merkt, dass die Library nicht auf allen Windows-95-Systemen vorhanden ist. Da die Fehlermeldung wiederum irreführend ist (s.o.), war der Fehler für die Entwickler zunächst schwer einzugrenzen.

■ Es gab sodann Dokumentationsfehler: Macromedia schlug in einer Technote zum Thema vor, dass die MSVCRT.DLL wie die anderen DLLs in den Projektor-Ordner gelegt werden sollte. Das war für Director 7 aber noch keine Lösung (in Director 8 ist es nun aber der richtige Weg). Nach dem Erscheinen von Director 7.02 wurde ein Installer für die MSVCRT.DLL und eine weitere Library (MSVFW32.DLL) auf der Macromedia-Website zur Verfügung gestellt (<http://download.macromedia.com/pub/director/technotes/>

[D7projector\\_dlls\\_installer.zip](http://www.macromedia.com/go/14151) und <http://www.macromedia.com/go/14151>). In Director 8 kann diese DLL im Temp-Verzeichnis (Standardprojektor, Komprimierter Projektor) oder im Projektor- bzw. im XTRAS-Verzeichnis des Projektors (Faststart-Projektoren) liegen.

■ Eine weitere DLL (MSVFW32.DLL) sorgte für Verwirrung; auch sie ist unter Windows 95 nicht in jedem Fall vorhanden (nämlich dann nicht, wenn der Anwender eine angepasste Installation durchgeführt und die Multimedia-Libraries nicht installiert hat). Der oben genannte Installer installiert auch diese Datei. In Director 8 ist diese Library nicht mehr zum Start des Projektors nötig. Ist sie nicht vorhanden, so werden lediglich AVI-Filme nicht angezeigt.

■ Auf dem Mac war es weder mit Director 7 noch mit Version 8 nötig, Faststart-Projektoren zu erstellen. Standardprojektoren haben die gleiche Ladezeit. Falls Sie in Director 8 dennoch einen Shockwave-Projektor mit externen Libraries erstellen wollen, beachten Sie, dass auf dem Mac nun auch vier Libraries nötig sind: PROJLIB, DPLIB, IMLLIB und MACROMEDIARUNTIMELIB.

■ PROJ.DLL und PROJLIB enthalten Code, der aus der Programmdatei des Projektors ausgegliedert wurde. Das Verfahren soll sicherstellen, dass Shockwave-Projektoren von heute mit zukünftigen Versionen von Shockwave funktionieren. Au-

ßerdem wurde die Größe der eigentlichen Programmdatei damit weiter reduziert: auf 12 KByte am Mac und 60 KByte unter Windows (statt 180 KByte zuvor auf beiden Plattformen).

## Hintergrundwissen zu Projektoren

Projektoren bestehen aus folgenden Bestandteilen:

1. Projektoranwendung
2. Eingebundene Player-Libraries (ca. 2,3 MByte)
  - DPLib, IMLLib, MacromediaRuntimeLib, ProjLib (unter MacOS)
  - dirapi.dll, iml32.dll, msvcrt.dll, proj.dll (unter Windows)
3. Eingebundene Xtras (komprimiert).
4. Eingebundene Medien (evtl. Shockwave-komprimiert)

Die Projektoranwendung (1.) hat in Director 8 ca. 12KByte unter MacOS und ca. 60KByte unter Windows. Ein Shockwave-Projektor bindet die Libraries (2.) nicht ein und ist daher sehr kompakt. Er ist aber darauf angewiesen, dass Shockwave 8 installiert ist.

Die Größe eines komprimierten Projektors ist geringer als die eines Standardprojektors, da die Libraries, die unter (2) aufgelistet sind, komprimiert werden. Dies reduziert die Dateigröße um ca. 1 MByte, bedeutet aber längere Startzeiten des Projektors, da die komprimierten Daten auch wieder entpackt werden müssen. Der Zeitbedarf liegt bei ca. fünf Sekunden auf einem durchschnittlichen Rechner, kann aber auch über 20 Sekunden betragen (beim Programmstart von Diskette etc.).

Ein Standardprojektor enthält die unkomprimierten Libraries. Die oben beschriebene Lösung, einen Faststart-Projektor zu erstellen, wirkt sich auf die Größe im Vergleich zum Standardprojektor kaum aus, da dieselben unkomprimierten Libraries verwendet werden.

Shockwave-Projektoren (ohne die entsprechenden Libraries) sind immer von einer installierten Shockwave-Version abhängig. In Director 7 musste exakt die richtige Version vorhanden

sein, damit ein Shockwave-Projektor funktionierte. Mit Director 8 ist der Projektor nur noch ein „Loader“ für die ProjLib, die mit Shockwave installiert wird. Dieser „Loader“ soll laut Macromedia auch in der Lage sein, spätere Shockwave-Installationen zu nutzen, so dass bis zu einer Änderung des DCR-Formats hier Anwender-Sicherheit herrscht (Macromedia stellt eine Rückwärtskompatibilität beim DCR-Format von mindestens zwei Versionen sicher.)

### Splash-Screen anzeigen



Nutzen Sie das folgende (meines Wissens undokumentierte) Verfahren, um in Director 8 einen Splash-Screen anzuzeigen: Erstellen Sie eine Pict- (Mac) bzw. BMP-Datei (Win), und benennen Sie sie genau wie Ihren Projektor (plus die Endung .PCT bzw. .BMP). Unter Windows lassen Sie die Dateiendung .exe des Projektors weg. Legen Sie die Datei in das gleiche Verzeichnis wie Ihren Projektor (Abbildung 9.21). Die Grafikdatei wird angezeigt, noch bevor die Bestandteile des Projektors geladen sind.

Eigene Tests ergaben allerdings, dass das Feature nicht wirklich verlässlich funktioniert. Wir hatten unter Windows 95 im Hintergrund startende Projektoren sowie auf allen Plattformen Situationen, in denen der Splash-Screen einfach nicht angezeigt wurde. Eigenes Testen ist also unerlässlich.

## 9.5 Die Ini-Dateien

### Lingo.ini, on startup

Im Director-Ordner finden Sie eine LINGO.INI genannte ASCII-Textdatei mit folgendem Inhalt (Abbildung 9.22).

Sowohl Director als auch ein Projektor suchen beim Start im aktuellen Ordner nach einer Datei mit der Bezeichnung LINGO.INI und führen die darin enthaltene Prozedur `on startup` noch vor dem Start der Wiedergabe des Films aus.



Abbildung 9.22: Lingo.ini im Texteditor

Damit eignet sich die Datei LINGO.INI hervorragend zum Debugging in Projektoren. Beispielsweise können Sie hier als Movie-in-a-Window laufende Debugging-Tools öffnen oder auch nur eine globale Variable setzen, die bestimmte Testausgaben im zu untersuchenden Hauptprogramm initiiert. Übrigens funktionieren `put`-Anweisungen – entgegen dem Kommentar in der INI-Datei – im Projektor und in der Windows-Authoring-Umgebung sehr wohl.

Auch während des Authorings ergeben sich sinnvolle Anwendungsmöglichkeiten. So erwarten einige Xtras – wie z.B. BudApi oder MasterApp – vor jeder Verwendung die Angabe einer Registrierungsnummer. Um diese nicht bei jedem Start von Director wiederholen zu müssen, können Sie die Registrierungsaufrufe einfach in die Datei LINGO.INI kopieren. Ebenso lassen sich lokale Suchpfade zu Projektverzeichnissen in der LINGO.INI setzen, was Ihnen die Eingabe von langen Pfadangaben beim Öffnen von Dateien über das Nachrichtenfenster erspart.

```

on startup
    the searchpaths = ["f:\projekte\Projekt1\"]
end

```

Listing 9.33: Suchpfade in Lingo.ini setzen

Dass Sie diese Version der LINGO.INI dann nicht mit Ihren Projektoren vertreiben sollten, versteht sich von selbst.

### Director.ini, Projektorname.ini, Shockwave.ini

Mit Director 8 hat Macromedia begonnen, die Datei DIRECTOR.INI, die sich im Director-Verzeichnis befindet, ausführlich zu kommentieren. Damit können Sie sich nun ein Bild von den meisten ihrer Funktionen machen.

Diese Ini-Datei kann an drei verschiedenen Orten zum Einsatz kommen: als DIRECTOR.INI im Director-Verzeichnis (für das Authoring); als PROJEKTORNAME.INI (im gleichen Verzeichnis wie ein Projektor, der PROJEKTORNAME bzw. PROJEKTORNAME.EXE heißt) oder als SHOCKWAVE.INI im Shockwave-Verzeichnis (unter MacOS ist das der Ordner MACINTOSH HD:SYSTEMORDNER:SYSTEMERWEITERUNGEN:MACROMEDIA:SHOCKWAVE 8, unter Windows das Verzeichnis C:\WINDOWS\SYSTEM\MACROMED\SHOCKWAVE 8\). Die jeweils möglichen Einstellungen sind unterschiedlich.

Die INI-Datei ist in dem von Windows bekannten Format aufgebaut, das so aussieht:

```
[Abschnittsbezeichnung1]
Propertybezeichnung1=Propertywert1
Propertybezeichnung2=Propertywert2
...
[Abschnittsbezeichnung2]
...
```

Zeilen werden mit einem Semikolon am Zeilenanfang auskommentiert. Wenn Sie eine Einstellung nutzen wollen, müssen Sie auch die entsprechende Abschnittsbezeichnung „aktivieren“ (also das Semikolon bei beiden entfernen). Statt TRUE und FALSE sind stets 1 und 0 zu verwenden, Leerzeichen sind nicht erlaubt (mit einer Ausnahme: siehe unten bei MessageWindowRect).

Die Abschnittsbezeichnungen sind:

```
[Palette]
[Settings]
[Lingo]
[Movies] *
[Memory] *
[Sound] *
```

*Listing 9.34: Struktur der Director.ini*

Die Abschnitte mit \* sind für MacOS nicht verfügbar. Der Abschnitt [Sound] erlaubt es, sehr spezielle Einstellungen für bestimmte Soundkarten unter Windows vorzunehmen und ist vor allem interessant, wenn Sie Probleme mit der Soundwiedergabe in der Authoring-Umgebung haben. Nähere Informationen hierzu finden Sie in den Kommentaren direkt in der INI-Datei. Von wenig praktischem Nutzen sind die Einstellungen im Abschnitt [Palette], da Abspielplattformen mit 8 Bit Farbtiefe doch eher der Vergangenheit angehören.

```
[Palette]
Animation=1
```

Die Einstellung beschleunigt Palettenübergänge, da Director vollständig die Kontrolle über die Systempalette übernimmt. Die Standardeinstellung ist 0 (Director darf die Kontrolle nicht übernehmen. Palettenübergänge werden verlangsamt, aber die Wahrscheinlichkeit von Palettenkonflikten wird reduziert).

```
SystemFriendly=1
```

Auf 8-Bit-Systemen werden Director-Paletten auf die Web-Standard-Palette umgerechnet (#web216 plus zehn Graustufen). Standardeinstellung ist 0 (mit Director-Farbwechseln ändern sich auch die Systemfarben). Die Einstellung wurde mit der Applikation Shockmachine (vgl. KAPITEL 7 SHOCKWAVE-INTERNET IM DIREKTZUGRIFF → 377) eingeführt, die stets „systemfreundlich“ betrieben wird.

### Überschreiben der Projekteinstellungen/ Projektordebugging

Die meisten Einstellungen, die im Dialogfeld PROJEKTOR ERSTELLEN/OPTIONEN vorgenommen werden, lassen sich über die Einstellungen im Abschnitt [Settings] überschreiben. Außerdem gibt es noch einige weitere Optionen, die insbesondere für das Testen und Debuggen sehr komfortabel sind.

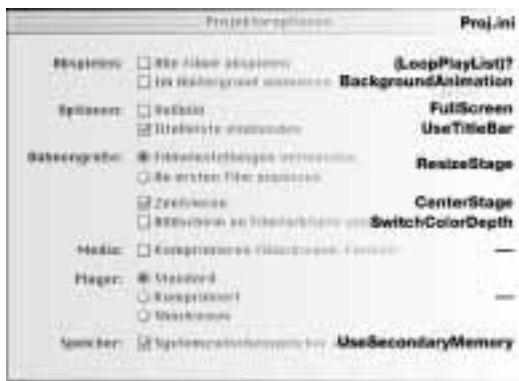


Abbildung 9.23: Projektoroptionen:  
Mit einer Projektorname.ini lassen sich  
die Voreinstellungen modifizieren.

Abbildung 9.23 zeigt, welche Einstellungen der Projektoroptionen durch welche Properties in der INI-Datei überschrieben werden können. Sie können also einen Projektor erstellen und in der Testphase über die PROJEKTORNAME.INI-Datei die voreingestellten Projektoroptionen verändern, um beispielsweise den Vollbild-Modus auszusuchen, ein Nachrichtenfenster anzuzeigen und einige Speicheroptionen auszuprobieren.

Tabelle 9.1 zeigt alle bekannten Einstellungen im Überblick.

Die Einstellungen SingleInstance und DisplayFullLingoErrorText sind ebenso wie die Speichereinstellungen nur über die PROJEKTORNAME.INI zu erreichen. Die meisten Einstellungen sind auch in Director 7.02 verfügbar; allerdings hatte es Macromedia versäumt, sie entsprechend zu dokumentieren.

Häufig wird die PROJEKTORNAME.INI genutzt, um beim Testen von Projektoren temporär Projekteinstellungen zu überschreiben, ein besseres Fehler-Reporting zu erhalten etc. Die Einstellung [Settings] SingleInstance=1 sollte allerdings zum Standard-Inventar jedes Windows-Projektors gehören, wird damit doch auf äußerst simple Weise ein grundlegendes Problem unter Windows behoben.

### Shockwave-Einstellungen verändern

Liegt die DIRECTOR.INI als SHOCKWAVE.INI im oben genannten Shockwave-Verzeichnis, so lassen sich die Einstellungen MessageWindow und DisplayFullLingoErrorText auch in Shockwave nutzen. Unter Windows wird dabei tatsächlich ein Nachrichtenfenster im Browser geöffnet; unter dem MacOS wird eine Ausgabedatei erzeugt, die die Nachrichten, die in das Nachrichtenfenster ausgegeben würden, aufnimmt. Sie finden sie im Shockwave-Verzeichnis unter dem Namen 1MSGDUMP.TXT.

### Xtrainfo.txt

XTRAINFO.TXT ist eine Konfigurationsdatei, die von Director genutzt wird, wenn ein neuer Film oder ein Projektor erstellt wird. Sie enthält die Information, welche Xtras standardmäßig einem Projektor hinzugefügt werden, welche Xtras über eine Internet-URL heruntergeladen werden können und für welche Xtras zusätzliche Informationen im Internet zur Verfügung stehen. Die Datei organisiert außerdem Subsets von Xtras, die über die Buttons „Netzwerk hinzufügen“ und „Standard hinzufügen“ im Menü MODIFIZIEREN/FILM/XTRAS... aufgerufen werden können. XTRAINFO.TXT stellt Director auch die unterschiedlichen Dateinamen für Xtras auf den Plattformen MacOS und Windows zur Verfügung.

| Projektname.ini                                    | Erläuterung  |
|--|--|
| [Settings]<br>MessageWindow=1                      | Öffnet das Nachrichtenfenster im Projektor (sichtbar nur bei FullScreen=0).  |
| MessageWindowRect=0 300 320 400                    | Platzierung des Nachrichtenfensters: Angaben genau wie hier – also nur mit Leerzeichen getrennt – formatieren.   |
| LoopPlayList=0                                     | <u>undokumentiert</u> : Entsprechung zu „Alle Filme abspielen“ in den Projektoroptionen.   |
| EscapeOk=0   | Schaltet alle Quit-Tasten aus (EscapeOk=0). Das entspricht der Lingo-Einstellung the exitlock = TRUE. Auch die Schließbox eines Fensters mit Titelleiste funktioniert mit dieser Einstellung nicht mehr. |
| FullScreen=0                                       | Die Entsprechung zu „Vollbild“ in den Projektoroptionen.   |
| UseTitleBar=1                                      | Die Entsprechung zu „Titelleiste einblenden“ in den Projektoroptionen. Auch möglich bei FullScreen=1.  |
| BackgroundAnimation=1                              | Die Entsprechung zu „Im Hintergrund animieren“ in den Projektoroptionen.   |
| SwitchColorDepth=0                                 | Die Entsprechung zu „Bildschirm an Filmfarbtiefe anpassen“ in den Projektoroptionen.   |
| CenterStage=1                                      | Die Entsprechung zu „Bühnengröße: Zentrieren“ in den Projektoroptionen.  |
| ResizeStage=0                                      | Die Entsprechung zu „Bühnengröße: Filmeinstellungen verwenden“ in den Projektoroptionen.   |
| SingleInstance=1                                   | <u>nur Windows</u> : Verhindert den Mehrfachstart von Projektoren.   |
| [Movies]<br>Movie01=Filmname1<br>Movie02=Filmname2 | <u>undokumentiert, nur Windows</u> : Erlaubt es, die beim Erstellen des Projektors integrierten Filme zu ersetzen und/oder ihre Reihenfolge zu verändern.  |
| [Memory]<br>ExtraMemory=400                        | <u>nur Windows</u> : Über den Arbeitsspeicher hinausgehender Speicherbereich in KByte, den ein Projektor nutzen kann.  |
| SwapFileMeg=0                                      | <u>nur Windows</u> : Festlegung des für den Projektor nutzbaren Bereichs der Auslagerungsdatei (Swap-File) in MByte.   |
| ExtraMainMemory=8000                               | <u>nur MacOS, nicht dokumentiert</u> : Funktion unklar.  |
| UseSecondaryMemory=0                               | <u>nur MacOS, nicht dokumentiert</u> : Möglicherweise die Entsprechung zu „Systemzwischenpeicher verwenden“ in den Projektoroptionen.  |
| ExtraSecondaryMemory=0                             | <u>nur MacOS, nicht dokumentiert</u> : Funktion unklar   |
| [Lingo]<br>DisplayFullLingoErrorText=1             | Erlaubt es, ausführliche Fehlermeldungen statt der für Projektoren üblichen Meldung „Skriptfehler. Weiter?“ zu erhalten.   |

Tabelle 9.2: Director.ini: für Projektoren nutzbare Einstellungen. Die Eigenschaften MessageWindow und DisplayFullLingoErrorText sind auch in einer Shockwave.ini wirksam.

Die Datei liegt im Verzeichnis der Director-Applikation; sie muss nicht mit dem Projektor mitgeliefert werden.

Sie können die Datei XTRAINFO.TXT editieren, wenn Sie eines der folgenden Ziele erreichen wollen:

- Ein Xtra eines Drittanbieters soll in einen Projektor eingeschlossen werden, den Sie erstellen.
- Sie wollen das Set der Xtras, die automatisch in einen Projektor eingeschlossen werden, selbst festlegen.

■ Sie wollen Xtras für den automatischen Download („Bei Bedarf herunterladen“) markieren, die diese Option bisher nicht bieten oder gar nicht in der Liste der verfügbaren Xtras auftauchen. (vgl. Kapitel 7 Shockwave, bzgl. Xtras, die nicht in der Shockwave-Standardinstallation vorhanden sind → 376)

Um ein Xtra für den automatischen Download in XTRAINFO.TXT einzutragen, benötigen Sie vom Hersteller des Xtras die URL, unter der das Download-Package zur Verfügung steht.

Ein Eintrag in die Datei XTRAINFO.TXT hat folgende Form:

```
[#namePPC: "SWA Decompression PPC Xtra", #nameW32:
"swadcmpr.x32", #type: #default, #package: "http://
download.macromedia.com/pub/shockwave/xtras/SWA/
SWA", #info: "http://www.macromedia.com/shockwave/
xtras/SWA/index.html"]
```

```
[#namePPC: "INetUrl PPC Xtra", #nameW32:
"INETURL.X32", #net:#xtra, #type:#default]
```

-- mindestens:

```
[#namePPC: "FileIO PPC Xtra", #nameW32:
"FILEIO.X32"]
```

#namePPC und #nameW32 bezeichnen die Dateinamen des Xtras auf den unterschiedlichen Plattformen. (In Director 6 waren hier noch weitere Properties nötig: #nameW16, #name68K, #nameFAT.) Der optionale Eintrag #type: #default sorgt dafür, dass das Xtra in jedem neuen Director-Film in der movieXtraList aufgeführt wird (und in einen Projektor eingeschlossen wird, wenn Sie die Einstellung nicht ändern). #net: xtra ist die Kennzeichnung des Sets von Xtras, das über den Button NETZWERK HINZUFÜGEN im Dialogfeld FILM-XTRAS (Abbildung 9.24) zum Film und zur movieXtraList hinzugefügt wird. Mit der Property #info kann die URL angegeben werden, die beim Klick auf den Button INFO im Dialogfeld aufgerufen wird. Die Property #package legt fest, dass das Xtra automatisch aus dem Internet heruntergeladen werden kann, und ent-

hält die Information, unter welcher URL das Download-Package zu finden ist.

Weitere Einträge in der Datei XTRAINFO.TXT sind Kommentare; sie beginnen stets mit einem Semikolon oder mit --.

### Neuen Eintrag erstellen

Wollen Sie Xtras eines Drittherstellers (wie beispielsweise das PDFXtra von Integration New Media) in die XTRAINFO.TXT aufnehmen, um das Xtra in einen Projektor integrieren zu können oder es für einen Auto-Download zu markieren, so erstellen Sie einen Eintrag wie den folgenden:

```
; PDF Xtra by Integration New Media, Inc.
[#nameW32: "pdf asset.x32", #namePPC: "pdf asset
ppc", #package: "http://www.integration.qc.ca/
products/pdfxtra/verisign/1.0.2/PDFasset", #info:
"http://www.integration.qc.ca/products/pdfxtra"]
```

Listing 9.35: Xtrainfo.txt: PDF-Xtra eintragen

Im Dialogfeld FILM-XTRAS können Sie dann über HINZUFÜGEN... das Xtra in die movieXtraList aufnehmen, und die Optionen „In Projektor einschließen“ und „Bei Bedarf herunterladen“ sind anwählbar (Abbildung 9.24). Lassen Sie sich vom Hersteller des Xtras immer die aktuelle URL für #package mitteilen.



Abbildung 9.24: Das PDF Asset Xtra wurde zur movieXtraList hinzugefügt.

### VDL-Dateien

Die View Descriptor Lists (VDL) befinden sich in den Unterordnern des Verzeichnisses PROPS im Direc-

tor-Verzeichnis. Sie legen fest, welche Eigenschaften in der Listenansicht (definiert in der Subliste `#propertyDescriptionList`) bzw. in der Dialogansicht (`#itemdescriptionList`) des Property-Inspektors angezeigt werden. Da der Property-Inspektor kontextsensitiv ist, ist eine ganze Reihe von VDLs für Film-, Cast-, Sprite- und Darstellereigenschaften zuständig. Auch die Dialogfelder des Publish-Dialogs (DATEI/VERÖFFENTLICHUNGSEINSTELLUNGEN) werden durch eine VDL beschrieben.

VDL-Dateien sind reine Textdateien, die Sie auch an Ihre eigenen Bedürfnisse anpassen können. Das ist für die Dialogansicht relativ komplex, da auch die Positionsinformationen der einzelnen Dialogelemente sowie Icons und Labels angegeben werden müssen. In der Listenansicht können dagegen einfach fehlende Properties nachgetragen oder unerwünschte entfernt werden. Wenn Sie eine VDL ändern, müssen Sie Director nicht neu starten, um sie zu nutzen. Drücken Sie einfach die `[⇧]`-Taste beim Öffnen des Property-Inspektors, um alle VDLs neu zu laden. Übrigens ist es sinnvoll, eine Kopie des unberührten PROPS-Verzeichnisses anzulegen, bevor Sie die folgenden Anregungen ausprobieren.

Hier einige Ideen, wo es sinnvoll sein könnte, die VDLs zu editieren:

### Presets für Bühnengröße ändern

Sie wollen eine eigene Stage-Größe im Aufklappenmenü des FILM-Tabs des Property-Inspektors unterbringen – wie in Abbildung 9.25 den Eintrag „400x400“?

In der Datei `PROPS/MOVIE/DEFAULT.TXT` sind dafür an zwei Stellen Änderungen vorzunehmen: Suchen Sie zunächst `#movieProps` -> `#itemdescriptionList` -> `#size`, und dort dann die Einträge `#range` und `#list`. In beiden Sublisten ist die neue Größe (als String für die Anzeige, als Punktwert für das interne Handling) einzugeben.

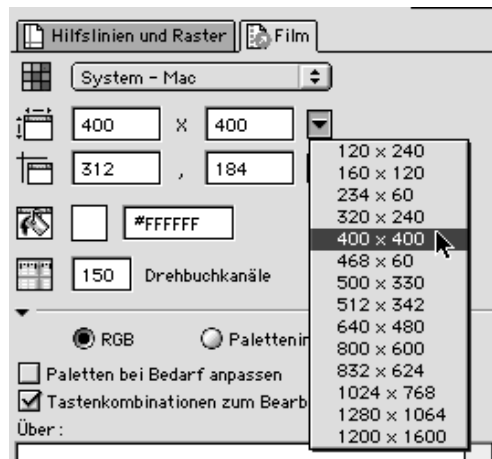


Abbildung 9.25: Neuer Eintrag für die Bühnengröße

```
-- Movie tab of Property Inspector
[
  #grid:
  (...)
  #movieProps:
  [
    #comment: "Film",
    #symbol: #movie,
    #helpTopic: 1103,
    #picture: [7900, 7],
    #propertyDescriptionList:
    [
  (...)
  #itemDescriptionList:
  [
  (...)
  #size:
  [
    #type: #menu,
    #range:
    [
      "120 x 240",
      "160 x 120",
      "234 x 60",
      "320 x 240",
      "400 x 400",
      "468 x 60",
```

```

"500 x 330",
"512 x 342",
"640 x 480",
"800 x 600",
"832 x 624",
"1024 x 768",
"1280 x 1064",
"1200 x 1600"
],
#list:
[
    Point(120,240),
    Point(160,120),
    Point(234,60),
    Point(320,240),
    Point(400,400),
    Point(468,60),
    Point(500,330),
    Point(512,342),
    Point(640,480),
    Point(800,600),
    Point(832,624),
    Point(1024,768),
    Point(1280,1064),
    Point(1600,1200)
],
#left: 160,
#top: 28,
#right: 160 + 14,
#bottom: 28 + 16
],
(...)

```

Listing 9.36: Props/Movie/Default.txt (Ausschnitt)

#### Vorauswahl für Kommentarfeld angeben

Für eine einfache Versionsverwaltung mag es sinnvoll sein, nur „strukturierte“ Eingaben in das Kommentarfeld des Property-Inspektors zu erlauben. In der Dialogansicht ist dies schwierig, da Sie sinnvollerweise auch eine ganze Reihe von Positionsangaben anderer Elemente ändern müssten. In der Listenansicht fügen Sie einfach eine

entsprechende #range in der Beschreibung der Property #comments ein:

```

#comments:
[
    #format: #string,
    #comment: "Kommentare (...)"
    , #range:
    [
        "3 - Abgenommen",
        "2 - OK",
        "1 - Bearbeitet",
        "0 - Status 0"
    ]
],

```

Listing 9.37: Props/Member/Default.txt (Ausschnitt)

Damit erzeugen Sie das kleine Aufklappmenü in Abbildung 9.26.

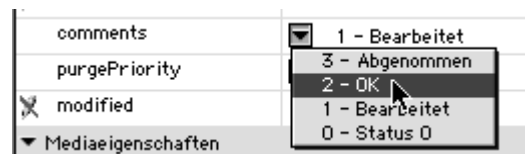


Abbildung 9.26: Vorgegebene Werte für das Kommentarfeld

#### Property-Angaben hinzufügen bzw. entfernen

Der Property-Inspektor enthält nicht alle Properties, die für die diversen Darstellertypen existieren. Wollen Sie beispielsweise die Properties translation und rotation im QUICKTIME-Tab des Inspektors „nachrüsten“, so gehen Sie folgendermaßen vor:

- Öffnen Sie die Datei PROPS/MEMBER/QUICKTIME-MEDIA.TXT.
- Fügen Sie an einer passenden Stelle (z.B. unter Anzeigeeigenschaften) die fett gedruckten Zeilen ein:

```

#propertyDescriptionList:
[
  #title:
  [
    #comment: "Abspieleigenschaften",
    #title: true
  ],
  (...)
  #title:
  [
    #comment: "Anzeigeeigenschaften",
    #title: true
  ],
  (...)
  #translation:
  [
    #format: #list,
    #comment: "Liste mit zwei Elementen. Gibt
die Verschiebung in x- und y-Richtung an."
  ],

  #rotation:
  [
    #format: #integer,
    #editable: true,
    #range:
    [
      #min: 0,
      #max: 360,
      #granularity: 1
    ],
    #comment: "Rotationswert des
Darstellers."
  ],
  (...)

```

Listing 9.38: Props/Member/quicktimemedia.txt (Ausschnitt)

Das Ergebnis sieht so aus wie in Abbildung 9.27.

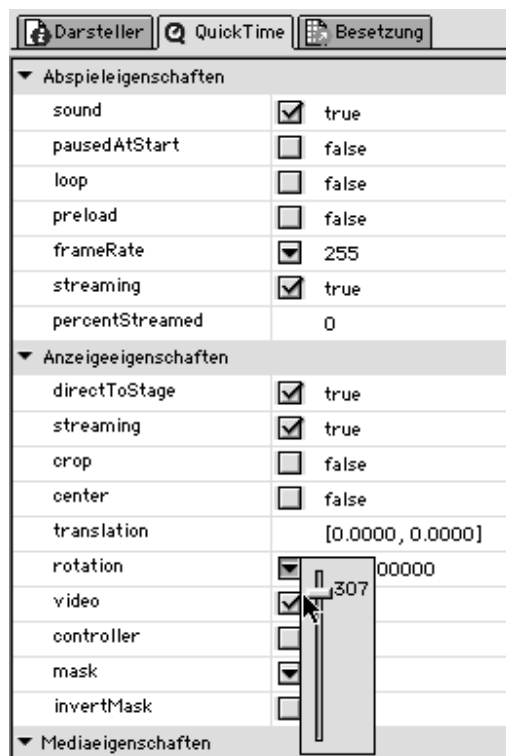


Abbildung 9.27: Die Darsteller-Eigenschaften translation und rotation im Property-Inspektor

## 9.6 Score-Recording

Mit Score-Recording können Sie die Inhalte des Drehbuchs skriptgesteuert ändern oder gar ein ganzes Drehbuch aufbauen. Was sich paradisi-sch anhört („Wow, dann läuft ja alles automa-tisch!“), ist in der Praxis aber nur sehr einge-schränkt nutzbar.

Es gibt Entwickler, die Score-Recording zur Laufzeit einsetzen. Der Sinn ist uns nie klar ge-worden – schließlich ist mit den üblichen Sprite-Manipulationen eine einfachere und umfassen-dere Kontrolle über das Drehbuch möglich.

## Drehbuchvorbereitung mit Score-Recording

Nützlich ist Score-Recording hingegen, wenn zur Drehbuchvorbereitung lästige Aufgaben automatisiert abgearbeitet werden können. Das folgende Skript legt im Abstand von zehn Frames Marker an und fügt ein Frame-Skript ein; außerdem legt es das der Markernummer entsprechende Bild aus der Castlib "dummys" in das Drehbuch. Wenn Sie mehrere Dutzend oder gar hunderte von Frames haben, die ansonsten manuell vorbereitet werden müssten, sollten Sie es einmal mit Score-Recording versuchen.

Es entsteht dabei ein Drehbuch, das als Grundlage zur Weiterarbeit dienen soll – wenn in der endgültigen Anwendung einfach nur Bilder ausgetauscht werden sollten, würde man wohl einen anderen Ansatz wählen.

Das Skript benötigt einen Darsteller "markers", aus dem die Marker-Bezeichnungen gelesen werden. Die zu platzierenden Elemente (Frame-Skript und Bilder) müssen natürlich auch vorhanden sein.

```
on preparescore
  mymarkers = member("markers").text
  nummarkers = mymarkers.paragraph.count()
  beginRecording
    repeat with i = 1 to nummarkers
      go to frame i*10
      labeltext = mymarkers.line[i]
      if labeltext <> "" then
        the frameLabel = labeltext
        --
        the frameScript = member("goframe")
        --
      repeat while TRUE
        sprite(2).member = member(i,"dummys")
        sprite(2).loc = point (400,300)
        updateFrame
        if the frame = (i+1)*10 then exit repeat
      end repeat
    end repeat
  end repeat
end repeat
end if
end repeat
endRecording
end
```

*Listing 9.39: Drehbuchvorbereitung mit Score-Recording (Filmskript)*

Die Lingo-Befehle `beginRecording` und `endRecording` umschließen die Score-Recording-Session. `updateFrame` trägt die Änderungen in das Drehbuch ein und lässt den Abspielkopf auf den nächsten Frame vorrücken. Das `go to frame i*10` positioniert den Abspielkopf vor jedem Recording-Abschnitt neu. In Abbildung 9.28 sehen Sie das so entstandene Drehbuch.

## 9.7 Häufigste Problemfelder

Manche Probleme in der Director-Authoring-Umgebung können auf bestimmte Einstellungen zurückzuführen sein, die Sie zuallererst überprüfen sollten:

### ■ Speicherprobleme am Mac, Crashes

Stellen Sie sicher, dass beim Authoring die Einstellung `SPEICHER: SYSTEMZWISCHENSPEICHER VERWENDEN` aktiviert ist.

### ■ Probleme mit Registrierungspunkten, generell Probleme im Malfenster

Stellen Sie sicher, dass die Einstellung „Zwiebelschichten“ (Menü `ANSICHT/ZWIEBELSCHICHTEN`, Button „Zwiebelschichteneffekt ein-/ausschalten“) ausgeschaltet ist.

### ■ Probleme mit Asset-Xtras (vor allem Windows)

Wenn Sie bestimmte Asset-Xtras (die einen eigenen Darstellertyp anlegen) und ActiveX-Controls nutzen, kann es nötig sein, die Einstellung „Im Hintergrund animieren“ im Menü `DATEI/VOREINSTELLUNGEN/ALLGEMEIN` einzuschalten. Diese Einstellung ist dann auch bei der Erstellung eines Projektors nötig (Menü `DATEI/PROJEKTOR ERSTELLEN`, Button „Optionen“).

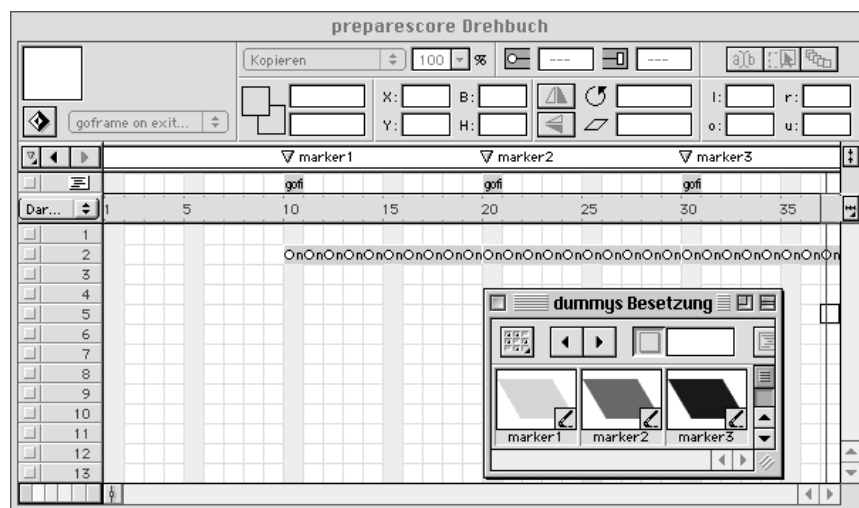


Abbildung 9.28: Rapid Prototyping: Marker, Frame-Skripte und Bildinhalte wurden per Lingo-Skript in das Drehbuch eingetragen.

Die folgenden Ausführungen nennen Ihnen weitere häufige Problemfelder und einige Lösungsmöglichkeiten. Zum Themenbereich Skriptfehler und Debugging von Skripten vgl. Kapitel 2 SKRIPTING-GRUNDLAGEN → 150.

### Probleme mit Farbpaletten

Vergleichen Sie allgemein zur Problematik der Farbpaletten Kapitel 1, ARBEITSTECHNIKEN → 51. Hier wollen wir die häufigsten Probleme mit Farbpaletten kurz aufzählen – in der Hoffnung allerdings, dass die Arbeit mit 8-Bit-Grafiken in Director der Vergangenheit angehört.

Die folgenden Probleme treten auf Systemen mit 8 Bit Farbtiefe zur Laufzeit auf:

- Schwarzweißer Bildschirm direkt nach dem Projektorstart

Dieses Problem ist in Director 5 und 6 unter Windows oft aufgetreten, wenn ungeduldige Anwender beim Start des Projektors auf den Desktop klickten. Es existierte vor Director 8 *kein wirklicher Workaround* in Director. Gewöhnlich schaltet man der Director-Applikation eine kleine Startapplikation vor, die den Anwender „bei der

Stange hält“ und die Wartezeit überbrückt. Auf der Director-6-CD befand sich eine solche Applikation (FASTSTART.EXE).

- Farb-Flash/Falschfarben beim Eintritt in einen Frame

Wenn Grafiken mit zwei unterschiedlichen Farbpaletten in einem Frame liegen, kann Director (bzw. der Projektor) dies in Maßen ausgleichen, wenn die Einstellung „Paletten bei Bedarf anpassen“ im Menü MODIFIZIEREN/FILM/EIGENSCHAFTEN... ausgewählt ist. Besser ist es, unterschiedliche Paletten strikt zu vermeiden, da die 8-Bit-Darstellung grundsätzlich von einer – bildschirmweiten – Farbpalette ausgeht.

- Falschfarben in der Desktop-Darstellung, Farb-Flashes beim Aufruf externer Applikationen, Farb-Flashes beim Aufruf von Dialogboxen, „verwirrte“ Director-Palette beim Aufruf externer Applikationen

Siehe oben. Externe Applikationen benutzen häufig andere Farbpaletten als Director. Beim Umschalten zwischen Programmen müssen auch die Paletten umgeschaltet werden, was zumeist nicht sehr ansehnlich ist. In Maßen (nämlich dann, wenn die externen Applikationen eben-

falls die 32 Standard-Windows-Farben auf den Palettenpositionen 0 bis 15 und 240 bis 255 nutzen) ist dem Farb-Flash-Problem durch die Verwendung der Windows-Systempalette in Director beizukommen. Das Problem der „verwirrten“ Director-Palette kann vermieden werden, wenn für den Aufruf einer externen Applikation kurzfristig auf einen Screen mit einer Schwarzweißdarstellung umgeschaltet wird; auch die – unsinnig aussehenden – Aufrufe `set the stageColor = the stageColor` oder `go frame (the frame) of movie (the movie)` haben schon geholfen, da sie einen Redraw der Bühne bzw. eine Neuinitialisierung des Filmes hervorrufen. In der Datei `DIRECTOR.INI` (bzw. `PROJEKTORNAME.INI`) kann außerdem folgende Einstellung vorgenommen werden, die Director davon abhält, allein die Kontrolle über die Palette zu übernehmen:

```
[Palette]
Animation=0
SystemFriendly=1
```

`SystemFriendly` bewirkt, dass *im Projektor* das Umschalten auf eine Nicht-Standard-Palette verhindert wird.

#### ■ Falschfarben im Video

Videos werden – auf beiden Plattformen – von der jeweiligen Systemsoftware (QuickTime oder Video for Windows) neu berechnet und in 8 Bit gedithert, wenn sie im Original eine höhere Farbtiefe haben. Bei Elementen, die eigentlich nahtlos in den Bühnenhintergrund übergehen, kann daher ein deutlicher Unterschied zur Director-Grafik beobachtet werden. *Workaround*: Es gibt Video-Codecs, die eine Kompression in 8 Bit mit festlegbarer Palette ermöglichen. Komprimieren Sie Ihr Material neu, und nutzen Sie die in Director verwendete Palette.

#### ■ „Banding“ – Streifenbildung in der Grafik

Schalten Sie für Grafiken mit mehr als 8 Bit Farbtiefe die Option `member(m).dither = 1` an (vgl. Kapitel 1, ARBEITSTECHNIKEN → 51). Ein in Director 7.0 bestehendes Problem mit „Banding“ in Quick-

TimeVR-Darstellern ist zwischenzeitlich behoben worden.

## Pfadprobleme

Director sollte (seit Version 7) für alle externen Darstellertypen (`#bitmap`, `#animGIF`, `#digitalVideo`, `#quickTimeMedia`, `#flash`, `#sound`, `#swa`), die als „gelinkte“ Darsteller in die Besetzung importiert wurden, in der Lage sein, relative Dateireferenzen aufrechtzuerhalten. Damit ist es möglich, eine Verzeichnisstruktur von der Festplatte beispielsweise auf CD-ROM zu übertragen.



**AKTUALISIEREN** (im Menü *XTRAS/FILME AKTUALISIEREN...*) wird von Macromedia als letzter Schritt vor der Distribution von Filmen empfohlen. Dieser Befehl – wie auch der Menübefehl *DATEI/KOMPAKT SICHERN* – bewirkt, dass die Datei neu geschrieben wird und als Nebeneffekt alle Referenzen auf externe Medien neu angelegt werden.

In Director 6.5 gab es hier vor allem mit dem neu eingeführten `QT3ASSETXTRA` Probleme, da es immer absolute Pfade speicherte. Dieses Problem ist seit Director 7 behoben (vgl. Technote 13746 im Director-Support-Center unter <http://www.macromedia.com/go/13746>).

Bedenken Sie beim Testen, dass Director bzw. der Projektor stets die Medien am Originalpfad nutzen wird, wenn diese vorhanden sind. Wenn Sie Ihre CD-R also auf dem Entwicklungsrechner testen, besteht die Möglichkeit, dass nicht die Daten von der CD, sondern Ihre lokalen Daten von der Festplatte genutzt werden.



Abbildung 9.29: Pfadfehler gehören zu den häufigsten Runtime-Problemen.

Pfadprobleme können entstehen, wenn

- die aufgerufenen externen Medien nicht existieren,
- Sie absolute Pfade statt relativer in Ihren Skripten verwenden,
- Sie Standardbenennungen voraussetzen (z.B. für das Windows-Verzeichnis bzw. den Systemordner), anstatt den FileIO-Xtra-Befehl `getOSDirectory()` zu nutzen,
- Pfadangaben nicht für die Plattform angepasst wurden, auf welcher der Film läuft,
- im SafePlayer-Modus mit bestimmten Lingo-Befehlen Dateien außerhalb eines DSWMEDIA-Verzeichnisses aufgerufen werden (vgl. Kapitel 7, SHOCKWAVE → 411), oder
- `the searchCurrentFolder = FALSE` gesetzt wurde, Dateien im aktuellen Ordner aber nur mit Dateinamen ohne vollständige Pfadangabe aufgerufen werden.

### Relative Pfade

Relative Dateiverweise sind eine unbedingte Voraussetzung für fast alle Aktionen, die Sie mit externen Dateien ausführen können. Sie haben dabei verschiedene Möglichkeiten, um relative Pfade zu konstruieren:

```
aPath = the moviePath & \
"video:einFilm.mov" -- bzw.
aPath = the moviePath & "video\einFilm.mov"
```

Dies würde einen Film adressieren, der z.B. an folgenden Orten liegen kann (der aktuelle Film liegt im Ordner MEDIA):

- C:\PROJEKTE\KUNDE1\WORK\MEDIA\VIDEO\EINFILM.MOV
- MACINTOSH HD:MACTEST:MEDIA:VIDEO:EINFILM.MOV
- oder auf der CD-ROM: E:\MEDIA\VIDEO\EINFILM.MOV bzw. CD-NAME:MEDIA:VIDEO:EINFILM.MOV

Angenommen, der zum Film gehörige Projektor liegt eine Ebene höher, also beispielsweise im Verzeichnis WORK, MACTEST oder im Root-Verzeichnis der CD-ROM. Derselbe Pfad könnte dann mit `the applicationPath` so angegeben werden:

```
aPath = the applicationPath & \
"media:video:einFilm.mov"
aPath = the applicationPath & \
"media\video\einFilm.mov"
```

Da sich die Pfade für MacOS und Windows immer durch den Pfad-Delimiter unterscheiden, können Sie sich mit einer kleinen Funktion viel Tipperei sparen:

```
on pd
  if the platform contains "Mac" then return ":"
  else return "\"
end
```

*Listing 9.40: Pathdelimiter 1 (Filmskript)*

Noch kürzer ist:

```
on pd
  return the last char of the applicationPath
end
```

*Listing 9.41: Pathdelimiter 2 (Filmskript)*

Damit wären die Pfadangaben von oben Plattform übergreifend so zu formulieren:

```
aPath = the applicationPath & "media" & pd()
&"video" & pd() & "einFilm.mov"
```

Auch mit dem Platzhalter @ können Sie relative Pfade konstruieren, die den Vorteil haben, dass sie Plattform übergreifend nutzbar sind; @ entspricht der Positionsangabe des `moviePath`.

```
aPath = "@/video/einFilm.mov"
```

Da allerdings nicht alle Lingo-Schnittstellen, die Pfadangaben benötigen, auch den Operator @ akzeptieren, ist der durchgängige Einsatz von @ nicht möglich. Ausgeschlossen bleiben beispielsweise die meisten Xtras und Shockwave.

### Die Suchfunktionen von Director

Dateien, die sich im selben Verzeichnis wie der Projektor (bzw. der gerade aktuelle Film) befinden, werden von Director automatisch gefunden,

wenn Sie sie einfach nur mit dem Dateinamen (ohne Pfadangabe) aufrufen. Die Systemproperty `the searchCurrentFolder` schaltet dieses Verhalten ein (Standard) oder aus. Einen ähnlichen Effekt können Sie erreichen, wenn Sie die Systemproperty `the searchPaths` (Director-6-Syntax: `the searchPath`) setzen.

```
put the searchPaths
-- []
add (the searchPaths, the moviePath & "video")
put the searchPaths
-- ["c:\projekte\kunde1\work\media\video"]
```

*Listing 9.42: the searchPaths (Nachrichtenfenster)*

Setzen Sie `the searchPaths`, bevor Director, der Projektor oder ein Lingo-Statement eine externe Datei verifiziert oder anfordert. Produziert Director beispielsweise beim Öffnen eines Filmes einen „File not found“-Fehler, so nützt die Angabe von `the searchPaths` in diesem Film nichts mehr. Setzen Sie die Property in einem vorangehenden Film, am besten im Startfilm Ihres Projekts oder in der `LINGO.INI` (vgl. *Die Ini-Dateien* in diesem Kapitel → 539).

`the searchPaths` kann auch genutzt werden, um zwischen unterschiedlichen Varianten Ihrer externen Medien umzuschalten.

Wenn Sie zahlreiche Verzeichnisse in `the searchPaths` eintragen, kann die Performance Ihrer Applikation darunter leiden.

## Speicherprobleme

Wenn sich das Tempo Ihrer Animationen mit zunehmender Dauer der Wiedergabe verlangsamt oder Director gar größere speicherintensive Darsteller komplett ausblendet, ist in der Regel ein „zugewucherter“ Arbeitsspeicher die Ursache. Das interne Speichermanagement von Director ist ebenso ausgefeilt wie effizient. Die Darsteller kommender Bilder werden automatisch vorausgeladen, und nicht mehr benötigte Darsteller werden automatisch wieder aus dem Speicher entfernt. Dies gilt ohne Einschränkung aller-

dings nur für linear ablaufende, drehbuchbasierte Filme.

Speicherprobleme erkennen Sie an folgenden Erscheinungen:

- Explizite Warnungen (z.B. beim Import zahlreicher Grafiken: „Nicht genügend Speicher...“) werden angezeigt.
- Audio spielt nicht oder hat Aussetzer
- Ganze Darsteller werden nicht angezeigt (besonders große Grafiken); integrierte Fonts werden nicht genutzt.
- Ständiger kurzzeitiger Festplattenzugriff (Director entlädt und lädt möglicherweise immer wieder den gleichen Darsteller.)
- Trails und Artefakte auf der Bühne, obwohl Trails („Spuren“) ausgeschaltet sind.
- Performance-Einbrüche treten auf, wenn die Anwendung auf Rechnern mit wenig Speicher abgespielt wird.
- Es kommt zu Crashes. Zumeist wird Director mit einer Warnung darauf hinweisen, dass der Film gesichert werden muss; es gibt aber auch Situationen, die das Programm (oder den Projektor) zum Absturz bringen. Eine ziemlich sichere Möglichkeit, einen Absturz wegen eines *stack overflow* zu erreichen, sind rekursive Handler wie in folgendem Beispiel:

```
on crashMe -- Warnung: Nomen est omen!
  crashMe
end
```

- „Handler not defined“-Fehler treten – unerklärlicherweise – auf.

Mit den folgenden einfachen Verfahren können Sie die häufige Speicherprobleme beseitigen:

- Setzen Sie am MacOS die Speicherzuteilung der Applikation auf einen höheren Wert bzw. schalten Sie (im Authoring) „Systemzwischenpeicher verwenden“ in den Voreinstellungen ein.
- Setzen Sie globale Variablen auf `VOID`, wenn Sie sie nicht mehr benötigen. Geben Sie Speicher frei, indem Sie Fenster richtig schließen (`forget`

window(w)) und löschen Sie alle Referenzen auf Objekt- und Xtra-Instanzen.

- Schließen Sie, wenn Sie einen Film mit `play movie` aufrufen, den Aufruf mit `play done` ab. Vermeiden Sie es, vom aufgerufenen Film aus weitere `play-` oder `go to movie-`Befehle aufzurufen. Der Film, der den ursprünglichen `play movie`-Befehl ausgegeben hat, wird weiter im Speicher gehalten. Normalerweise ist `go to movie` vorzuziehen, da dann der aufrufende Film entladen wird.

- Daten, die mit `importFileInto` in einen Film importiert werden, belegen Arbeitsspeicher, bis der Film gesichert wird. Vor allem im Projektor ist es sinnvoll, auf `importFileInto` ganz zu verzichten und statt dessen `member(m).filename` oder `url` zu setzen. Der gleiche Sachverhalt gilt auch für dynamisch generierte Darsteller (`new(#bitmap)` etc.) und damit ganz besonders für Imaging Lingo (vgl. Kapitel 3 LINGO FÜR ANIMATION UND BILDMANIPULATION → 192). Auch diese Darsteller müssen im RAM gehalten werden, bis Sie sie explizit wieder löschen.

- Große Sounds sollten als externe Dateien mit `sound playFile` abgespielt werden. Diese Methode lädt sie nicht vor und hält sie nicht weiter im Speicher. Interne Sounds werden ganz vorgeladen und verbleiben im RAM.

- Gehen Sie intelligent mit Grafikdarstellern und Text um! Es gibt zahlreiche Möglichkeiten, Speichererfordernisse zu reduzieren. Vgl. Kapitel 1 ARBEITSTECHNIKEN.

### Entlade-Reihenfolge

Wenn Lingo die Bildfolge bestimmt – und das ist eher die Regel als die Ausnahme – ist es sinnvoll, die automatische Speicherverwaltung von Director mit den entsprechenden Lingo-Befehlen zum Vorausladen und Löschen von Darsteller- und/oder Bildfolgen zu unterstützen. Darüber hinaus kann im Dialog des Befehls `DARSTELLERINFO` für die meisten Darsteller angegeben werden, mit wel-

cher Priorität sie aus dem Speicher entfernt werden sollen (the `purgePriority` of `member`). Allerdings besagt diese Einstellung nur, dass Darsteller später als andere gelöscht werden sollen. Sie ist damit z.B. für Button-Rollover-Grafiken geeignet, die in der ganzen Anwendung vorkommen: Hier ist es sinnvoll, die Grafiken möglichst nicht ständig zu entladen und wieder neu zu laden.

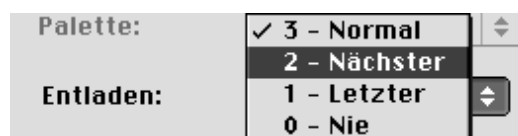


Abbildung 9.30: Die Option „Entladen“ im Dialog „Darstellerinfo“

### Das Diashow-Problem

Wenn Sie Darsteller durch das Setzen der Property `sprite(s).member` auf die Bühne bringen, behält Director den ursprünglich verwendeten Darsteller im Speicher. Wenn Sie sicher sind, dass der Darsteller erst einmal nicht mehr benötigt wird, ist das explizite Entladen mit dem Befehl `unload member` zu empfehlen. Nur so ist sichergestellt, dass der Wechsel des mit einem Darsteller verbundenen Sprites den zur Verfügung stehenden Arbeitsspeicher nicht negativ beeinflusst.

Das folgende Behavior illustriert den Unterschied. Erzeugen Sie z. B. in den ersten 128 Positionen der Besetzung größere Bitmaps, und weisen Sie einem beliebigen Sprite das Behavior zu. Es weist dem Sprite nachfolgend die Darsteller 1 bis 128 zu und gibt nach jedem Wechsel des Darstellers den freien Speicher in KByte aus.

```
property i
on beginsprite me
  i = 0
  put the freeBytes / 1024
end
```

```

on enterframe me
    i = i + 1
    if i > 128 then i = 1
    sprite(me.spritenum).member = member(i)
    updateStage
    put the freeBytes / 1024
end
    
```

Listing 9.43: Speichertest, Variante 1 (Behavior-Skript)

Variante zwei speichert vor jedem Wechsel des Darstellers den ursprünglich mit dem Sprite verknüpften Darsteller in der lokalen Variablen `oldmem` und entlädt ihn nach der Umschaltung des Darstellers. Der `enterFrame`-Handler ersetzt den `enterFrame`-Handler von Variante 1:

```

on enterframe me
    i = i + 1
    if i > 128 then i = 1
    oldmem = sprite(me.spritenum).member
    sprite(me.spritenum).member = member(i)
    updateStage
    unload oldmem
    put the freeBytes / 1024
end
    
```

Listing 9.44: Speichertest, Variante 2 (Behavior-Skript)

Im folgenden Diagramm ist der Verlauf des freien RAM in Abhängigkeit von der Zahl der geladenen Darsteller aufgetragen. Beim expliziten `unload` bleibt der freie Speicher praktisch konstant. Überlässt man Director die Speicherverwaltung, so werden zunächst alle Darsteller im Speicher behalten; erst wenn der freie Speicher knapp wird, finden `unload`-Vorgänge statt (Spitzen in der unteren Linie).

### unload im Drehbuch

Eine sehr einfache Möglichkeit, bei Bildfolgen, die im Drehbuch ausgelegt sind, ein manuelles `unload` immer genau dann durchzuführen, wenn die Anzeige eines Darstellers beendet ist, bietet das folgende Behavior, das einfach auf jedes in Frage kommende Sprite gezogen wird:

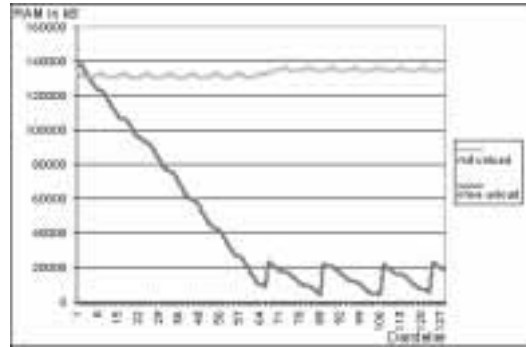


Abbildung 9.31: Der Verlauf des freien Arbeitsspeichers mit und ohne manuelle Entladung

```

on endSprite me
    unload sprite(me.spritenum).member
end
    
```

Listing 9.45: Sprites entladen (Behavior-Skript)

Dieses Vorgehen ist beispielsweise dann sinnvoll, wenn Sie über lange Strecken grafikintensive Seiten präsentieren, die der Anwender durchblättern kann. Das `unload` von wenigen Darstellern geht relativ schnell vonstatten und verhindert ebenso wie die Lösung im obigen Abschnitt ein „Zulaufen“ des Speichers.

Pauschale `unload`-Befehle wie beispielsweise

```

unload
unload 1, the frame - 1
    
```

können zu Entladevorgängen im Sekundenbereich führen, was für die meisten Anwendungen inakzeptabel lang ist.

### Manuelles Purge beim Authoring

Der Speicherinspektor (Menübefehl FENSTER/INSPEKTOREN/SPEICHER) leistet beim Authoring nur eingeschränkte Dienste bei der Klärung der Frage, wofür Director den Arbeitsspeicher gerade verwendet. Eine missverständlicherweise „Löschen“ (Purge) genannte Option bietet die Möglichkeit, den Arbeitsspeicher von Altlasten zu befreien (natürlich ohne Ihre möglicherweise ungesicherte Arbeit in Director zu löschen).

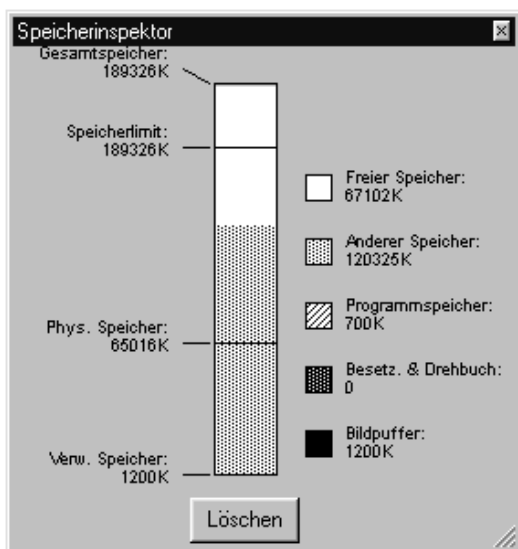
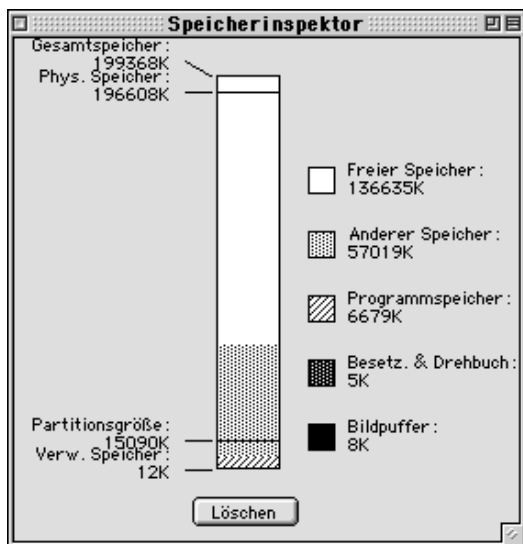


Abbildung 9.32: Der Speicherinspektor zeigt die Belegung des Arbeitsspeichers

Beachten Sie, dass Director beim Authoring deutlich mehr Speicher benötigt als im Projektor. Ins-

besondere wenn Sie Besetzungsfenster öffnen und scrollen, bemerken Sie, dass Director die Überblicksdarstellungen generiert und dafür Speicher verbraucht. In Director 7 und 8 kommt hinzu, dass für Textdarsteller, deren Option `pre-renderer` aktiviert wurde, bereits frühzeitig eine Bitmap angelegt wird, die – je nach Größe des Darstellers – eine beträchtliche Menge an Arbeitsspeicher belegen kann.

### Mac: Systemzweischenspeicher verwenden

**Speicher:**  Systemzweischenspeicher verwenden

Abbildung 9.33: Die Möglichkeit, Systemzweischenspeicher zu verwenden, ist im Authoring-Modus zumeist sinnvoll.

Ist die Option „Speicher: Systemzweischenspeicher verwenden“ im Menü `DATEI/EINSTELLUNGEN/ALLGEMEIN...` aktiviert, so ist Director beim Authoring in der Lage, über die normale RAM-Partition hinaus Systempeicher zu verwenden. Beim Authoring ist das im Allgemeinen eine sinnvolle Option.

Bei der Projektorerstellung gibt es eine gleichlautende Option (Menü `DATEI/PROJEKTOR ERSTELLEN/OPTIONEN`). Sie zu aktivieren bietet Ihnen die Möglichkeit, dem Projektor (fast) unbegrenzten Zugriff auf den Systempeicher des Macs zu geben. Wenn Sie die Option deaktivieren und sinnvolle Einstellungen im Speicherinfo des Projektors vornehmen (vgl. → Abbildung 9.34), haben Sie in den meisten Fällen aber Vorteile – z.B. wenn Sie parallel zum Projektor externe Anwendungen starten wollen oder wenn auf dem Abspielrechner virtueller Speicher aktiviert ist. Mehr Speicher macht einen Projektor nicht schneller. Effektives Entladen ist der bessere Weg.



Abbildung 9.34: Information/Speicher: Die Einstellungen im MacOS-Finder. In Director 7 waren hier unsinnig niedrige Werte voreingestellt. Director-8-Projektoren sind dagegen mit der Standardeinstellung im Normalfall nutzbar.

## Performance-Probleme

Performance-Probleme können heutzutage in zwei Richtungen auftreten: die Anwendung kann zu langsam, stockend oder fehlerhaft laufen – möglicherweise aufgrund mangelnder Performance des Abspielrechners oder vermeidbarer Performance-Killer in der Anwendung –, oder sie kann zu schnell laufen! Das zweite Problem war mir selbst auch neu – aber die Zeiten, als Entwicklungsrechner den Consumer-Geräten an Performance meilenweit voraus waren, sind vorbei.

Das zweite Problem lässt sich auch gut beobachten, wenn Sie alte Director-5-Produktionen auf einem modernen Rechner öffnen. Techniken, die besonders anfällig sind für Performance-Probleme dieser Art, sind:

### ■ idle-Aktionen

idle-Events werden in Director ab Version 7 häufiger ausgeführt als in früheren Versionen; mit the idleHandlerPeriod = 1 stellen Sie Director-6-Verhältnisse her, d.h. Director sendet maximal 60 idle-Events pro Sekunde

### ■ repeat-Loops

Animationen werden möglicherweise viel zu schnell abgespielt.

### ■ Fehlende Begrenzung der Bildrate

Eine solche Begrenzung war früher aufgrund langsamer Laufwerke und wenig RAM scheinbar nicht nötig.

Häufiger ist dagegen das erste Problem, das meistens dreistufig ist: Auf dem Entwicklungsrechner läuft die Anwendung wunderbar – wohl gemerkt: von Festplatte. Stufe 1 sind die Fehler, die man machen kann, um auch unter Idealbedingungen eine Anwendung in die Knie zu zwingen. Stufe 2 tritt in Kraft, wenn die Produktion von CD-ROM abgespielt werden soll, Stufe 3, wenn nicht der Abspielrechner, sondern das untere Ende der Zielplattform(en) ins Visier genommen wird.



Die Authoring-Umgebung selbst bremsst unter Umständen Ihre Anwendung erheblich: Insbesondere der Property-Inspektor und die Miniaturen-Ansicht des Besetzungsfensters können negative Auswirkungen auf die Abspielgeschwindigkeit haben, wenn dort ein Darsteller angezeigt wird, dessen Inhalt sich verändert. Das ist besonders häufig bei Text- und Bitmapdarstellern der Fall. Wenn Sie also unerklärliche Performance-Einbrüche beim Authoring haben, schließen Sie diese beiden Fenster.

Ein zusätzliches Performance-Problem, das nur auf dem Authoring-Rechner auftritt, wurde mehrfach von Director-Entwicklern berichtet: Wenn Director im Hintergrund geöffnet ist, laufen Projektoren und Shockwave-Filme in Director 7 deutlich langsamer. Dieser Fall tritt natürlich immer auf, wenn Sie mit dem Menübefehl VORSCHAU IM BROWSER einen Film im Browser starten. Das Problem sollte in Version 8 behoben sein.

### Stufe 1: Performance-Killer

■ Vermeiden Sie globale unload-Aufrufe. Das Entladen eines vollen Arbeitsspeichers kann mehrere Sekunden dauern. Meist sind gezielte unload member-Aufrufe effektiver (siehe Abschnitt Speicherprobleme → 551)

■ In einem Frame, auf dem Sie die Anwendung loopen lassen (go loop, go the frame etc.), sollten Sie bestimmte Dinge vermeiden. Stellen Sie sich

einfach vor, was passiert, wenn Director in einer Schleife mit einer Bildrate von 30 fps (Bilder pro Sekunde) Folgendes tun soll:

- eine Transition ausführen
- unload-Befehle ausführen
- komplexe `exitFrame`-, `prepareFrame`- oder `enterFrame`-Skripte ausführen

In diese Event-Handler gehören nur die Aktionen, die wirklich bei jedem Bildwechsel auszuführen sind. Die Initialisierung von Variablen, Properties etc. sowie alle Berechnungen, die nur einmal ausgeführt werden müssen, sollten Sie in den `beginSprite`-Handler schreiben.

- Vermeiden Sie `repeat`-Loops für Animationen. Es gibt bessere Techniken (vgl. Kapitel 3 ANIMATION UND BILDMANIPULATION MIT LINGO → 176).
- Halten Sie Ihre Director-Filme überschaubar. Kleinere Filme sind einfacher zu debuggen, und sie sind häufig performanter.
- Text- und Fontdarsteller sind unter Umständen für Performance-Einbussen verantwortlich. Vgl. Kapitel 1 ARBEITSTECHNIKEN → 57.

#### Probleme beim CD-ROM-Einsatz

- Halten Sie den Projektor klein. Sie können Startzeiten signifikant verbessern, wenn Sie
  - Xtras nicht in den Projektor packen lassen, sondern in einen separaten Xtras-Ordner neben Ihren Projektor legen,
  - nötige Programm-Libraries in das Verzeichnis der Projektor-Applikation (bzw. – mit Director 8 – in den Xtras-Ordner) legen und
  - nur einen Minimalfilm in den Projektor integrieren (vgl. Abschnitt *Projektoren* → 537).
- Vermeiden Sie, Video oder externen Sound vorzuladen.

■ Vermeiden Sie, auf mehrere Dateien gleichzeitig zuzugreifen. Beispielsweise wird ein QuickTime-Film anfangen zu stottern, wenn parallel ein Sound oder eine große Grafik geladen wird. Ebenso werden zwei gleichzeitig mit `sound play`-File abgespielte Sounds von CD-ROM Probleme machen.

■ Komprimieren Sie Ihr Video so, dass es vom „schlechtesten“ CD-ROM-Laufwerk spielen kann.

|              |                          |                      |
|--------------|--------------------------|----------------------|
| 2fach CD-ROM | 300 KByte/s<br>Datenrate | 200 KByte/s<br>Video |
| 6fach CD-ROM | 900 KByte/s<br>Datenrate | 600 KByte/s<br>Video |

#### Probleme bei „schwachen“ Rechnern

- Komprimierte Filme und Besetzungen (`dcr`, `cct`) benötigen mehr Rechnerleistung als geschützte Dateien (`dxr`, `cxt`).
- Große Bühnengröße, hohe Farbtiefe, Dither-, Alpha-, Blend- und Ink-Effekte erhöhen die Datenmenge, die zu laden ist, und/oder die erforderliche Rechenleistung für die Darstellung (vgl. Kapitel 1 ARBEITSTECHNIKEN → 103).
- Video, das nicht `direct-to-stage` abgespielt wird, benötigt mehr Rechnerleistung.
- Skalierung und Rotation werden zur Laufzeit berechnet und benötigen Rechnerleistung.
- Soundmixing wird unter Windows deutliche Verzögerungen aufweisen, wenn nicht DirectX 5 oder eine neuere Version installiert ist (Director 7.02 und 8 profitieren automatisch vom Soundmixing via `DirectSound`; vgl. Kapitel 4 VIDEO UND SOUND → 277).