

**SIEMENS**

**ANIMATOR™**

Testprogramm für LEVEL II COBOL™

**SINIX**

für Siemens PC

---

**ANIMATOR™ (SINIX)**  
**Testprogramm**  
**für LEVEL II COBOL™**

**Ausgabe Mai 1985 (ANIMATOR Version 1.0B)**

---

Bestell-Nr. U2222-J-Z95-1  
Printed in the Federal Republic of Germany  
800 AG 5852. (1000)

ANIMATOR und LEVEL II COBOL sind Warenzeichen  
von Micro Focus.

Alle Rechte vorbehalten.

Vervielfältigung dieser Unterlage sowie Verwertung ihres  
Inhalts unzulässig, soweit nicht ausdrücklich zugestanden.

Im Laufe der Entwicklung des Produktes können aus  
technischen oder wirtschaftlichen Gründen Leistungsmerk-  
male hinzugefügt bzw. geändert werden oder entfallen.  
Entsprechendes gilt für andere Angaben in dieser Druck-  
schrift.

Siemens Aktiengesellschaft

---

## Vorwort

Dieses Handbuch beschreibt **ANIMATOR** im Rahmen des Programmierpakets

- Übersetzer LEVEL II COBOL (LII COBOL)
- Laufzeitsystem LZS
- FORMS-2
- ANIMATOR

### Was enthält dieses Handbuch?

Dieses Handbuch beschreibt die Bedienung des ANIMATOR. Der ANIMATOR ermöglicht es Ihnen, ein mit dem LEVEL II COBOL-Übersetzer erzeugtes Programm direkt am Bildschirm, sagen wir, zum Leben zu erwecken.

Sie können nach eigenem Wunsch Teile des Programms mit ausgewählter Geschwindigkeit ablaufen lassen und den Inhalt von Datenfeldern an Unterbrechungspunkten oder während des Ablaufs überwachen. Sie können Ihr Programm prüfen und austesten, indem Sie nur das Quellprogramm verwenden, ohne Einzelheiten des erzeugten Zwischencodes zu kennen.

### Welche Handbücher gibt es noch für das LII COBOL Entwicklungssystem?

Neben der vorliegenden Bedienungsanleitung sind im Rahmen des LII COBOL Entwicklungssystems für die Siemens-PC noch folgende Handbücher erhältlich:

- LEVEL II COBOL Sprachbeschreibung
- FORMS-2 Bedienungsanleitung
- LEVEL II COBOL Bedienungsanleitung

### An wen wendet sich dies Handbuch?

Dieses Handbuch ist für LII COBOL-Programmierer bestimmt, die ihre Programme unter SINIX testen und ablaufen lassen.

---

## Wie ist das Handbuch aufgebaut?

- Kapitel 1 gibt eine allgemeine Einführung in den ANIMATOR. Hier finden Sie auch ein Beispiel, mit dem Sie Ihre ersten Schritte mit ANIMATOR an einem Testprogramm unternehmen können. Schon da werden Sie sehen, wie angenehm das Testen von Programmen sein kann!
- Kapitel 2 zeigt Ihnen, was beim Übersetzen mit ANIMATOR passiert und wie Sie den ANIMATOR starten.
- Kapitel 3 beschreibt Ihnen ganz ausführlich alle ANIMATOR-Kommandos in der Reihenfolge der Menüs.
- Kapitel 4 führt alle ANIMATOR-Kommandos in alphabetischer Reihenfolge auf.
- Kapitel 5 zeigt kurz und bündig alle ANIMATOR-Kommandos. Die Kommandos sind ganz knapp beschrieben, und stehen auf einem Blatt, damit Sie es aus dem Handbuch nehmen und beim Testen neben sich legen können.

## Wie finden Sie sich in diesem Handbuch zurecht?

Oben auf jeder Seite steht ein Hinweis, was in dem Abschnitt behandelt wird. Bei den ausführlich beschriebenen Kommandos im Kapitel 3 steht außen am Rand das Symbol für das Kommando und innen die englische Bedeutung.

## Was können Sie tun, damit das Manual besser wird?

Eine ganze Menge! Schreiben Sie uns bitte, an welchen Stellen das Handbuch falsch ist oder unübersichtlich - kurz, wo Sie im Handbuch gestolpert sind.

Oder schreiben Sie uns auch, wenn Sie mit dem Manual zufrieden sind, das freut uns dann ganz besonders!

Auf jeden Fall - dafür schon jetzt vielen Dank!

**Manualredaktion K D ST PM 2**  
Otto-Hahn-Ring 6, 8 München 83

---

# Inhalt

Seite

<b>1</b>	<b>Einführung</b>	<b>1-1</b>
1.1	Allgemeine Beschreibung	1-1
	Was können Sie mit ANIMATOR alles tun?	1-1
1.2	ANIMATOR Installieren	1-4
1.3	ANIMATOR ausprobieren	1-5
1.3.1	Übersetzen des Testprogramms stock1.CBL	1-5
1.3.2	ANIMATOR starten	1-6
1.4	Was kann der ANIMATOR nicht?	1-9
<b>2</b>	<b>Der Ablauf mit ANIMATOR</b>	<b>2-1</b>
2.1	Vorbereitungen	2-1
2.2	Ablauf	2-2
<b>3</b>	<b>Kommandos</b>	<b>3-1</b>
3.1	Allgemeines	3-1
3.2	Hauptkommandos	3-5
3.3	Fenster verändern	3-5
	<input type="checkbox"/> S Screen	3-6
	<input type="checkbox"/> S Screen Divider	3-7
	<input type="checkbox"/> F Find	3-8
	<input type="checkbox"/> O <input type="checkbox"/> A IOcAte	3-9
3.4	Ablauf- und Animationskontrolle	3-10
	<input type="checkbox"/> B break-point	3-10
	<input type="checkbox"/> T unTil	3-12
	<input type="checkbox"/> E Execute	3-13
	<input type="checkbox"/> L Level	3-15
	<input type="checkbox"/> P Program Counter	3-16
	<input type="checkbox"/> N Name	3-17
3.5	Überprüfen und Ändern von Daten	3-18
	<input type="checkbox"/> D Display	3-18
	<input type="checkbox"/> Q Query	3-19
	<input type="checkbox"/> M Monitor	3-20
3.6	Bildschirmanzeige des Benutzerprogramms	3-21
	<input type="checkbox"/> U User	3-21
<b>4</b>	<b>Die Kommandos in alphabetischer Reihenfolge</b>	<b>4-1</b>
<b>5</b>	<b>Kommandos, kurz und bündig</b>	<b>5-1</b>



# 1 Einführung

## 1.1 Allgemeine Beschreibung

### Was können Sie mit ANIMATOR alles tun?

Sie können

- Teile oder das ganze Programm beim Test mit einer Geschwindigkeit ablaufen lassen, die Sie wählen
- den Inhalt von Datenfeldern an vorgegebenen Unterbrechungspunkten oder spontan während des Ablaufs überwachen oder verändern
- den Ablauf Ihres Programms austesten, ohne den erzeugten Zwischencode zu kennen.

### Was will denn der ANIMATOR hauptsächlich?

ANIMATOR will Sie als LII COBOL-Programmierer davon befreien, den Inhalt des Datenbereichs und des Befehlsbereichs Ihres COBOL-Programms im Kopf zu haben. Dadurch ist selbst ein ungeübter Programmierer schon bald in der Lage, sein Programm effektiv auszutesten.

Das erreicht ANIMATOR dadurch, daß der Bildschirm als "Fenster" zum COBOL-Quellprogramm benutzt wird. Die "Animation" des Programms wird dadurch bewirkt, daß der Cursor im angezeigten COBOL-Programm von Befehl zu Befehl springt, genauso wie das Programm momentan ausgeführt wird. Die Ausführungsgeschwindigkeit kann dabei von Ihnen verändert werden. Sie können die Animation auch ausschalten, sodaß ein schnellerer Ablauf bis zu einem gewünschten Programmteil erreicht wird.

Sie können den Ablauf an jedem beliebigen Punkt unterbrechen, entweder durch Definition von Unterbrechungspunkten (break-points) oder dynamisch, indem Sie einfach die Leertaste drücken. Solange der Ablauf unterbrochen ist, können Sie sich den gewünschten Teil Ihres Programms am Bildschirm anzeigen lassen. Es ist also nicht einmal eine Übersetzungsliste notwendig, um das Programm fehlerfrei zu bekommen.



## Einführung

---

Ihnen stehen eine Menge Testfunktionen zur Verfügung, die Sie mit jeweils einer einzigen Taste auslösen. Dabei wird das Quellprogramm nur in den ersten 20 Zeilen des Bildschirms angezeigt; die verbleibenden Zeilen dienen dazu, das Auswahlménü für die einzelnen Kommandos bzw. deren Untermenüs anzuzeigen.

Beziehen sich Testfunktionen auf ein bestimmtes Datenfeld bzw. auf einen bestimmten Befehl des Programms, so bewegen Sie den Cursor auf das entsprechende Feld im Quellprogramm, um den Bezug der Funktion zum Datenfeld herzustellen. Datenfelder können auch durch Eingabe des COBOL-Datennamens angesprochen werden.

Wenn Sie bei einem Kommando mehr tun müssen, als den Cursor zu setzen oder einfach ein Kommando einzugeben (z.B. ein Literal angeben), dann müssen Sie das immer in der LII COBOL-Syntax tun.

### *Beispiel*

Sollen Inhalte von Datenfeldern verändert werden, geschieht dies durch Eingabe des Wertes im Format eines COBOL-Literals, d.h. nichtnumerische Literale sind in Anführungszeichen einzuschließen.

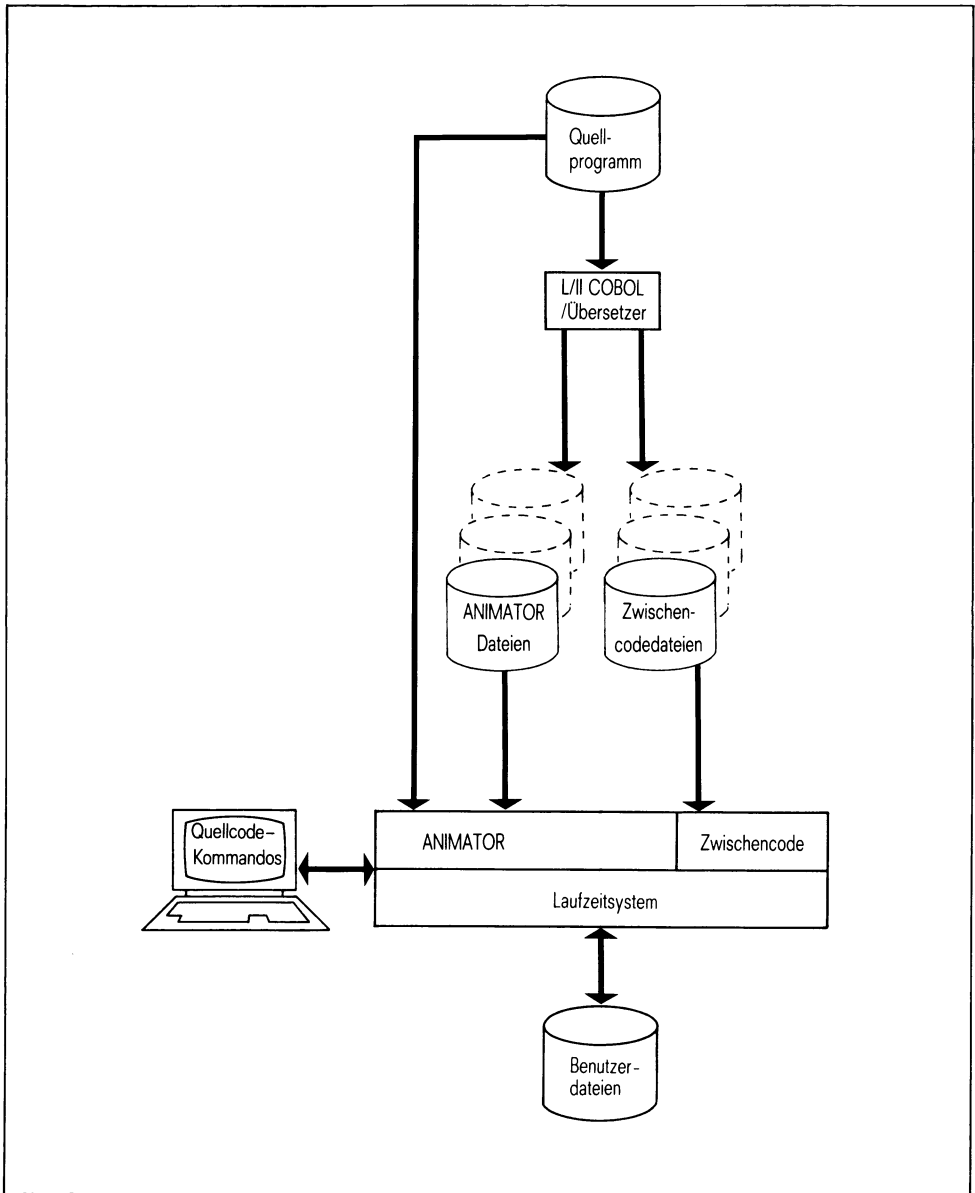


Bild 1-1 Stellung des ANIMATOR im LII COBOL System

### 1.2 ANIMATOR Installieren

Sämtliche für den ANIMATOR notwendigen Dateien befinden sich auf der Diskette COBANI, die Sie unter der Kennung von "admin" als Softwarepaket in Ihrem System installieren müssen (siehe auch LII COBOL Bedienungsanleitung). Sobald die Installation erfolgreich abgeschlossen ist, können Sie bereits Ihre ersten Schritte mit dem ANIMATOR versuchen. Dafür können Sie das mitgelieferte Programm stock1.CBL verwenden. Sie finden das Programm in dem Dateiverzeichnis /usr/lib/cobol/demo.

### 1.3 ANIMATOR ausprobieren

#### 1.3.1 Übersetzen des Testprogramms stock1.CBL

Vielleicht kennen Sie das Programm schon. Beim Installieren von LII COBOL diente es bereits als Demonstrationsprogramm.

Um das Demonstrations-Programm für den Ablauf mit ANIMATOR zu übersetzen, geben Sie folgendes Kommando ein:

```
cobol stock1.CBL anim reseq ref 
```

Hier sehen Sie: Damit Sie das Programm mit ANIMATOR ablaufen lassen können, müssen Sie beim Übersetzen **anim** angeben.

Der COBOL-Übersetzer gibt die gewohnten Start-Meldungen aus. In der Datei stock1.LST wird ein durchnummeriertes Übersetzungslisting abgelegt, das Sie sich gegebenenfalls ausdrucken lassen können, obwohl das Listing beim Testen mit dem ANIMATOR weniger wichtig ist als bei den üblichen Testhilfen. Das werden Sie aber selbst sehr schnell merken.

## ANIMATOR starten

---

### 1.3.2 ANIMATOR starten

Um den ANIMATOR zu starten, geben Sie folgendes ein:

```
animate stock1 
```

Der ANIMATOR meldet sich mit:

- \* Level II COBOL V 2.0 Copyright(c) 1982
- \* Accepted -RNIM

Dann erscheint das COBOL-Quellprogrammlisting am Bildschirm. Es beginnt mit der ersten Zeile der PROCEDURE DIVISION, dort blinkt der Cursor unter dem ersten Befehl.

Die zur Verfügung stehenden Kommandos werden unterhalb des Listings angezeigt. Sie können nun das Programm stock1 mit  für EXEC und  für GO animieren.

Das Programm läuft nun mit Geschwindigkeit 5 ab (Voreinstellung). Wollen Sie die Geschwindigkeit ändern, geben Sie eine Ziffer von 1 bis 9 ein, dabei ist 1 ganz langsam und 9 ganz schnell. Soll der Programmablauf unterbrochen werden, drücken Sie die Leertaste!

Um Ihnen ein Gefühl für die Arbeit mit dem ANIMATOR zu geben, folgen nun einige willkürliche Kommandos. Sie können natürlich jedes beliebige Kommando aus dem Kapitel 3 eingeben.

Kapitel 3 beschreibt alle Kommandos und Fehlermeldungen im Detail. Sie können die Kommandos groß oder klein geschrieben eingeben.

Geben Sie  für das Find-Kommando ein, dann wird die Kommandozeile durch eine Leerzeile ersetzt. Geben Sie nun z.B. "CRT-STOCK-CODE" ein (die Anführungszeichen sind unbedingt nötig) und drücken Sie die  Taste. ANIMATOR zeigt Ihnen dann, wo das erste Mal im Quellcode der Datenname auftritt (Zeile 000470).

Geben Sie **[M]** ein, um dieses Datenfeld zu überwachen. Die Möglichkeiten dieser Monitor-Funktion werden angezeigt. Drücken Sie **[S]**, um die Funktion einzuschalten. Eine Zeile mit # -Zeichen leuchtet kurz am Bildschirm auf; das bedeutet: ein Fehler. Aha, der Cursor steht am Ende des Datenfeldnamens. Um die Monitor-Funktion einzuschalten, muß der Cursor jedoch auf dem **ersten** Buchstaben des Datenfeldnamens stehen, das überwacht werden soll. Springen Sie mit dem Cursor an die richtige Stelle, indem Sie die **[←]** Taste drücken. Geben Sie nun wieder **[M]** und **[S]** ein. Es geschieht nichts, da das Feld derzeit ohne Inhalt ist. Sobald das Feld einen Wert bekommt, wird er unterhalb der Kommandozeile angezeigt.

Geben Sie **[B]** für break-point ein, um einen Unterbrechungspunkt zu definieren (d.h. einen Punkt, an dem das Programm immer anhält, sobald er durchlaufen wird).

Die Möglichkeiten für einen Unterbrechungspunkt werden angezeigt. Auch diesmal erkennen wir einen Irrtum: Unterbrechungspunkte können nur bei COBOL-Befehlen gesetzt werden. Diesmal drücken Sie die Leertaste, um zur Hauptkommandozeile zurückzukommen. Bewegen Sie den Cursor hinunter zum Wort MOVE in Zeile 00490 und setzen Sie ihn unter das M von MOVE.

Geben Sie nun wieder **[B]** und anschließend **[S]** ein, um hier einen Unterbrechungspunkt zu definieren.

Geben Sie nun **[E]** ein für das Ausführungskommando EXEC, und die Ausführungsmöglichkeiten werden angezeigt. Drücken Sie **[G]** für GO und das Programm wird mit der vorgegebenen Geschwindigkeit (5) ausgeführt. Geben Sie nun **[1]** ein, um die langsamste Ausführungsgeschwindigkeit auszuwählen, damit Sie genau verfolgen können, was geschieht. Der Cursor läuft nun durch den angezeigten COBOL-Code entsprechend der momentanen Ausführung. So wird das COBOL-Programm quasi zum Leben erweckt oder animiert.

Sie können die Animation jederzeit anhalten, indem Sie die Leertaste drücken; oder starten, wenn Sie die Taste **[G]** drücken. Beachten Sie, daß Sie das **[G]** eingeben können, ohne vorher **[E]** eingeben zu müssen.

Sie können auch nach **[E]** dann **[X]** für single step angeben; das Programm führt dann immer den nächsten Befehl aus, wenn Sie ein **[X]** eingetippen.

## ANIMATOR Beispiel

---

Die Dateneingabemaske wird angezeigt und Sie geben Werte ein, wie Sie es beim normalen Ablauf des Programms auch tun.

Die Daten werden entgegengenommen und das Programm hält an dem von Ihnen gesetzten Unterbrechungspunkt an (Zeile 00490). Beachten Sie, daß der von Ihnen im Feld STOCK-CODE der Eingabemaske eingegebene Wert nun unterhalb der Kommandozeile angezeigt wird, da Sie die Monitorfunktion für dieses Feld eingeschaltet haben.

Geben Sie wieder  für GO ein und wählen Sie nun die Geschwindigkeit 9. Beachten Sie den Unterschied der Animationsgeschwindigkeit. Geben Sie einen neuen Satz in die Dateneingabemaske des Programms ein und das Programm wird wieder am Unterbrechungspunkt anhalten und den neuen Wert des überwachten Datenfeldes unterhalb der Kommandozeile anzeigen.

Geben Sie  für ZOOM ein. Beachten Sie, daß Sie dieses Kommando eingeben können, ohne vorher  einzugeben. Die Animation wird ausgeschaltet und das Programm geht zu einer normalen Ausführung über. Geben Sie im Feld STOCK-CODE Blanks ein, und das Programm beendet sich. Nun können Sie das Ganze wiederholen, indem Sie wieder

animate stock1

eingeben. Probieren Sie nun andere Kommandos aus, die im Kapitel 3 stehen. Experimentieren Sie, und Sie werden sehen, es macht Spaß, mit so einem komfortablen Testsystem zu arbeiten!

Wollen Sie einen Überblick über alle Kommandos, sehen Sie im Kapitel 4 oder 5 nach!

### 1.4 Was kann der ANIMATOR nicht?

#### ANSI COBOL DEBUG Sprachmodul

Das COBOL Debug Sprachmodul sieht Möglichkeiten vor, Programmteile zu überwachen, indem Programmcode durch Laufzeit-Schalter aktiviert wird. Unter ANIMATOR ist das Aktivieren des Laufzeit-Schalters verboten, deshalb darf dieser Schalter nicht gesetzt werden.

#### Kommandozeilen-Parameter

LII COBOL bietet die Möglichkeit, dem Programm in der Aufrufzeile (cbrun programmname) Parameter mitzugeben. Diese Parameter werden von der ersten ACCEPT-Anweisung im ANSI-Format gelesen. Wenn ein Programm mit ANIMATOR läuft, geht das leider nicht. Solche Parameter müssen eingegeben werden, wenn der ACCEPT-Befehl durchlaufen wird.





## 2 Der Ablauf mit ANIMATOR

### 2.1 Vorbereitungen

Die zum Übersetzen eines Programms nötigen Schritte sind in der LII COBOL Bedienungsanleitung im einzelnen beschrieben.

Wollen Sie das Programm unter ANIMATOR ablaufen lassen, müssen Sie die Option "ANIM" angeben. ANIM können Sie groß oder klein schreiben.

Zwei andere Übersetzer-Funktionen können besonders nützlich sein: COPYLIST und RESEQ. Wenn Sie diese Optionen angeben, stimmt das Übersetzungslisting mit dem vom ANIMATOR am Bildschirm angezeigten Quellprogramm überein.

Für den Ablauf mit ANIMATOR erstellt der Übersetzer zusätzlich zur Zwischencode-Datei noch andere Dateien. Sie enden auf .ANM, .ACP, .DOO und, wenn das Programm segmentiert ist, auf .Dnn, wobei nn die Segmentnummer bedeutet. Diese Dateien, sowie das Quellprogramm selbst und alle COPY-Dateien müssen während des Ablaufs unter ANIMATOR vorhanden sein. Haben Sie ein Programm mit ANIM übersetzt, besteht übrigens kein Unterschied im Ablauf, ob Sie nun das Programm mit oder ohne ANIMATOR ablaufen lassen.

## 2.2 Ablauf

Um ANIMATOR aufzurufen, geben Sie das animate-Kommando ein.

Format:

animate programmname

Damit ANIMATOR arbeiten kann, müssen alle vorher erwähnten Dateien vorhanden sein.

### *Anmerkung*

Wenn Sie wünschen, daß Ihr Programm normal abläuft, bis ein spezielles Unterprogramm erreicht wird, müssen Sie folgende Optionen angeben:

animate programmname ZOOM \(\unterprogrammname\)

## 3 Kommandos

### 3.1 Allgemeines

Ist ein Programm zur Animation geladen, so wie es in Kapitel 2 beschrieben wurde, erscheint auf dem Bildschirm der Anfang des Befehlsbereichs. Der Cursor blinkt unter der ersten auszuführenden Anweisung. Die Ausführung wird durch eines der Kommandos, die in diesem Kapitel beschrieben werden, gestartet.

Am unteren Bildschirmrand (Zeile 23) folgt, von einer gestrichelten Linie getrennt, der erste Teil des Haupt-Kommando-Menüs:

```
000380 PROCEDURE DIVISION.  
000390 SR1.  
000400     DISPLAY SPACE.  
000410     OPEN I-O STOCK-FILE.  
000420     DISPLAY SCREEN-HEADINGS.  
000430 NORMAL-INPUT.  
000440     MOVE SPACE TO ENTER-IT.  
000450     DISPLAY ENTER-IT.  
000460 CORRECT-ERROR.  
000470     ACCEPT ENTER-IT.  
000480     IF CRT-STOCK-CODE = SPACE GO TO END-IT.  
000490     IF CRT-UNIT-SIZE NOT NUMERIC GO TO CORRECT-ERROR.  
000500     MOVE CRT-PROD-DESC TO PRODUCT-DESC.  
000510     MOVE CRT-UNIT-SIZE TO UNIT-SIZE.  
000520     MOVE CRT-STOCK-CODE TO STOCK-CODE.  
000530     WRITE STOCK-ITEM; INVALID GO TO CORRECT-ERROR.  
000540     GO TO NORMAL-INPUT.  
000550 END-IT.  
000560     CLOSE STOCK-FILE.  
000570     DISPLAY SPACE.  
-----  
B(rk-pnts) D(isp) E(xec) F(ind) L(evel) M(on) P(-c) Q(uey) S(creen) U(ser) ?
```

## Hauptkommando-Menü 2

---

Wenn Sie jetzt  eingeben, erhalten Sie den zweiten Teil des Hauptkommando-Menüs:

```
000380 PROCEDURE DIVISION.  
000390 SR1.  
000400     DISPLAY SPACE.  
000410     OPEN I-O STOCK-FILE.  
000420     DISPLAY SCREEN-HEADINGS.  
000430 NORMAL-INPUT.  
000440     MOVE SPACE TO ENTER-IT.  
000450     DISPLAY ENTER-IT.  
000460 CORRECT-ERROR.  
000470     ACCEPT ENTER-IT.  
000480     IF CRT-STOCK-CODE = SPACE GO TO END-IT.  
000490     IF CRT-UNIT-SIZE NOT NUMERIC GO TO CORRECT-ERROR.  
000500     MOVE CRT-PROD-DESC TO PRODUCT-DESC.  
000510     MOVE CRT-UNIT-SIZE TO UNIT-SIZE.  
000520     MOVE CRT-STOCK-CODE TO STOCK-CODE.  
000530     WRITE STOCK-ITEM; INVALID GO TO CORRECT-ERROR.  
000540     GO TO NORMAL-INPUT.  
000550 END-IT.  
000560     CLOSE STOCK-FILE.  
000570     DISPLAY SPACE.
```

-----  
N(ame) unT(il) locA(te) ?

Drücken Sie jetzt wieder , erscheint der erste Teil am Bildschirm.

Sie können folgende Funktionen vom Hauptmenü aus starten:

<b>B</b> (Break-point)	Unterbrechungspunkt setzen
<b>D</b> (Display)	Datenfelder anschauen
<b>E</b> (Exec)	ANIMATOR laufen lassen
<b>F</b> (Find)	Datenfeld suchen
<b>L</b> (Level)	Schachtelungstiefe bei Perform begrenzen
<b>M</b> (Monitor)	Monitorfunktion einschalten
<b>P</b> (Program counter)	Programmzähler erfragen oder setzen
<b>Q</b> (Query)	Datenfeld abfragen oder setzen
<b>S</b> (Screen)	Fenster verändern
<b>U</b> (User)	Benutzerbildschirm anzeigen
<b>N</b> (Name)	Programme zur Animation auswählen
<b>T</b> (unTil)	Unterbrechungsbedingung definieren
<b>O</b> (IOcAte)	Datendefinition suchen
<b>A</b>	
<b>?</b>	Der andere Teil des Hauptmenüs soll erscheinen

## Kommandos

---

Die Eingabe eines dieser Zeichen führt sofort dazu, daß die gewünschte Funktion ausgeführt wird, z.B., drücken Sie die Taste **F**, erscheint eine leere Kommandozeile, in die Sie den Namen Ihres Datenfeldes eintragen, das gesucht werden soll.

Eine ungültige Eingabe zeigt ANIMATOR an, indem er kurzfristig die Kommandozeile durch #-Zeichen ersetzt und piepst.

Einige der Haupt-Kommandos dienen nur dazu, untergeordnete Menüs mit detaillierteren Kommandos aufzurufen. Einige der häufiger verwendeten Unterkommandos (z.B. das **G** von vorhin) können auch direkt eingegeben werden, während das Hauptmenü angezeigt wird (siehe ausführliche Beschreibung dazu).

Nach Ausführung eines untergeordneten Kommandos erscheint wieder das Hauptmenü.

Bei einigen Kommandos muß der Cursor erst auf die entsprechende Position im angezeigten Quellprogramm gebracht werden.

Das Hauptmenü erscheint jedesmal, wenn die Ausführung unterbrochen ist. Solange dieses Hauptmenü angezeigt wird, kann der Cursor mit den normalen Cursor-Tasten über den Bildschirm geführt werden. Genauere Angaben dazu finden Sie in der LEVEL II COBOL Bedienungsanleitung.

Beachten Sie, daß Sie mit den Tasten **↑** und **↓** den Cursor innerhalb einer Spalte nach oben bzw nach unten führen können. Wenn Sie die Taste **↓** drücken, springt der Cursor an den Anfang der nächsten Zeile.

## 3.2 Hauptkommandos

Die Hauptkommandos lassen sich in vier große Bereiche einteilen:

- Quellprogramm "Fenster"-Bearbeitung mit **S** **F** **O** **A**
- Ablauf- und Animations-Kontrolle mit **B** **T** **E** **L** **P** **N**
- Überprüfen und Ändern von Daten mit **D** **Q** **M**
- Bildschirmanzeige des Benutzerprogramms mit **U**

Diese Kommandos lassen sich eingeben, wenn das Hauptmenü angezeigt ist.

## 3.3 Fenster verändern

ANIMATOR verwendet den Bildschirm als Fenster zum Quellprogrammtext. Mit den Kommandos können Sie sich einen beliebigen Teil des Quellprogramms am Bildschirm zeigen lassen.

ANIMATOR zeigt im Quellprogramm eine Zeilennummerierung an. Es ist die gleiche, die in der Übersetzungsliste erscheint, falls die wahlweisen Optionen RESEQ und COPYLIST bei der Übersetzung angegeben wurden.



**S** Screen

Wenn Sie **S** drücken, wird das folgende Untermenü angezeigt:

SCREEN - **N**(ext) **P**(revious) **T**(op) **E**(nd) **V**(iew) **H**(alf) **F**(ull) = / + /

Diese Kommandos verändern das Fenster bzw. verschieben den angezeigten Quellprogrammtext.

Sie haben folgende Möglichkeiten:

- N** zeigt die nächste Seite des Quellprogramms an. Beachten Sie, daß sich die Seiten um zwei Zeilen überlappen. Die gleiche Funktion erreichen Sie vom Hauptmenü mit der Taste **>**.
- P** zeigt die vorhergehende Seite des Quellprogramms an. Die gleiche Funktion erreichen Sie vom Hauptmenü mit der Taste **<**.
- T** zeigt Ihnen den Anfang des Quellprogramms.
- E** zeigt das Ende des Quellprogramms.
- V** verschiebt das Quellprogramm so, daß die Zeile, in der der Cursor steht, dann die dritte Zeile ist.

*Anmerkung*

Der Cursor muß bereits auf die gewünschte Zeile positioniert sein, bevor Sie das Kommando S des Hauptmenüs geben.

H

teilt den Bildschirm in zwei Fenster, die durch eine gestrichelte Linie getrennt sind. Im unteren Fenster sehen Sie den Anfang des Quellprogramms.

*Anmerkung*

Nachfolgende Bildschirmkommandos beziehen sich immer auf das Fenster, in dem der Cursor steht.

F

bewirkt die Rückkehr zu einem einzelnen Fenster.

=n

positioniert das Fenster so, daß die Quellprogrammzeile n als dritte Bildschirmzeile erscheint. n ist die am Bildschirm angezeigte Zeilennummerierung (ohne die führenden Nullen).

+n

schiebt das Quellprogramm um n Zeilen weiter vorwärts.

-n

schiebt das Fenster im Quellprogramm um n Zeilen zurück.

**S** **Screen Divider**

Dies ist ein Spezialfall des S-Kommandos:

Ist die Bildschirmanzeige geteilt (zwei Fenster) und der Cursor steht auf der gestrichelten Trennungslinie, dann wird nach **S** folgendes Untermenü angezeigt:

**SCREEN DIVIDER - U(p) D(own)**

Mit diesen beiden Kommandos können Sie die Größe der beiden Fenster verändern.

**U** verschiebt die Fenstertrennlinie eine Zeile nach oben.

**D** verschiebt die Fenstertrennlinie eine Zeile nach unten.

**F Find**

Dieses Kommando gibt dem ANIMATOR den Befehl, von der bestehenden Cursorposition aus eine bestimmte Zeichenfolge zu suchen. Ist sie gefunden, dann wird das Fenster so gesetzt, daß die Zeile mit der gesuchten Zeichenfolge als dritte Bildschirmzeile erscheint. Der Cursor steht am Ende der Zeichenfolge.

Falls die Zeichenfolge nicht gefunden wird, bleibt die Bildschirmanzeige unverändert und in der Kommandozeile wird das Hauptmenü angezeigt. ANIMATOR meldet durch eine Zeile mit lauter # und Piepsen, wenn er den gesuchten String nicht gefunden hat.

Nach dem F-Kommando sind folgende Eingaben möglich:

Format 1: "zeichenfolge"  (Überall suchen)

Format 2: "zeichenfolge" M  (nur im Hauptprogramm suchen und nicht in den COPY-Elementen)

- Anstelle von Anführungszeichen (") kann jedes beliebige Zeichen, das nicht in der gesuchten Zeichenfolge vorkommt, verwendet werden.
- Als Suchzeichenfolge ist jede beliebige Folge von Zeichen (incl. Zwischenraum) erlaubt.
- Die zusätzliche Eingabe von M veranlaßt den ANIMATOR, nur im Hauptprogramm nach der Zeichenfolge zu suchen (d.h. nicht in den COPY-Dateien).
- ANIMATOR durchsucht den Quellprogrammtext nur von Spalte 7 bis 72. Die angezeigte Zeilennummerierung wird beim Suchen der Zeichenfolge nicht berücksichtigt.

**O A IOcAte**

Mit diesen Kommandos können Sie entweder die Definition eines Datennamens oder eines Paragraphennamens im Quellprogramm suchen.

Sie haben zwei Möglichkeiten:

1. Sie setzen den Cursor an eine Stelle des Programms, an der der gesuchte Name steht und geben das Kommando **O** ein,
2. Sie geben zuerst das Kommando **A** und anschließend den gesuchten Namen ein, gefolgt von **J**.

Das Fenster wird nun so positioniert, daß der Cursor auf die Definition des gesuchten Namens zeigt.

### 3.4 Ablauf- und Animationskontrolle

Mit diesen Kommandos können Sie die Programmausführung starten und kontrollieren, aber auch wie schnell und bis zu welcher Schachtelungstiefe die Animation erfolgen soll.

#### **B** break-point

Nach Eingabe von **B** erscheint folgendes Untermenü:

BREAK-POINTS - S(et) U(nset) C(ancel) eX(amine) I(f)

Diese Kommandos definieren Unterbrechungspunkte (break-points), an denen das Programm bei der Ausführung stehen bleibt. Ein Unterbrechungspunkt kann mit einer Bedingung verknüpft sein.

Bei manchen Kommandos (die sind dann angegeben) muß der Cursor auf dem entsprechenden COBOL-Befehl stehen, d.h., der Cursor muß auf dem **ersten** Buchstaben des COBOL-Wortes stehen, ehe die Taste **B** gedrückt wird.

Bis zu vier Unterbrechungspunkte können gleichzeitig definiert werden.

Die Kommandos im einzelnen:

**S** definiert einen Unterbrechungspunkt an dem Befehl, auf dem der Cursor steht.

**U** hebt einen Unterbrechungspunkt an dem Befehl wieder auf, an dem der Cursor steht.

**C** löscht alle Unterbrechungspunkte.

**X** zeigt alle Unterbrechungspunkte an, indem es das Fenster des Quellprogramms verschiebt und den Cursor auf den entsprechenden Befehl setzt. Bei wiederholtem Aufruf dieser Funktion bewegt sich der Cursor von einem Unterbrechungspunkt zum anderen.

- I setzt einen Unterbrechungspunkt mit einer Bedingung bei dem Befehl, auf dem der Cursor steht. ANIMATOR positioniert den Cursor für die erforderliche Eingabe der Bedingung im COBOL-Format in die Kommando-Zeile. Die Eingabe wird durch  beendet.
- Fragt man einen Unterbrechungspunkt mit  ab, werden sowohl der Unterbrechungspunkt als auch die damit verbundene Bedingung angezeigt. Erreicht die Programmausführung diesen Unterbrechungspunkt, so hält sie nur an, wenn die Bedingung erfüllt ist.

**T unTil**

Mit diesem Kommando können Sie eine Bedingung definieren, die die Programmausführung jederzeit unterbricht, wenn die Bedingung wahr wird. Nachdem die Programmausführung unterbrochen wurde, wird die Bedingung nicht mehr abgefragt.

*Anmerkung*

Diese Funktion setzt die Ausführungsgeschwindigkeit beträchtlich herab.

Nach **T** wird folgendes Untermenü angezeigt:

UNTIL CONDITION - S(et) U(nset) eX(amine)

Sie haben folgende Möglichkeiten:

- S** setzt den Cursor in die Eingabezeile, in der Sie die geforderte Bedingung im COBOL-Format eingeben können.
- U** löscht eine vorher gegebene Bedingung.
- X** zeigt die gesetzte Bedingung an.



**E Execute**

Nach **E** wird folgendes Untermenü angezeigt:

EXECUTE - X(single step) sK(ip) l(till next If) G(o) Z(oom) S(top run)

Jetzt können Sie unter einer ganzen Menge Kommandos auswählen. Das einfachste Kommando hatten wir schon, das GO.

Aber jetzt die Kommandos im einzelnen:

- X** führt einen COBOL-Befehl aus und bewegt den Cursor auf den nächsten Befehl.
- K** überspringt einen COBOL-Befehl ohne ihn auszuführen, und setzt den Cursor auf den nächsten Befehl.

*Anmerkung*

Wenn der letzte Befehl eines Paragraphen übersprungen wird, der mit einer PERFORM-Anweisung aufgerufen wurde, wird das Programm nicht hinter dem PERFORM fortgesetzt, sondern an dem Befehl, der hinter dem aufgerufenen Paragraphen steht.

- I** Die Ausführung des Programms erfolgt ohne Animation bis zur nächsten If-Anweisung. An ihr wird die Programmausführung angehalten und der Cursor unter die If-Anweisung gesetzt.
- G** startet die Programmausführung mit Animation. Sobald ein Befehl ausgeführt worden ist, springt der Cursor zum nächsten auszuführenden Befehl im Quellprogramm. Die Geschwindigkeit der Programmausführung kann ständig während des Ablaufs geändert werden, wenn Sie eine Zahl zwischen 1 und 9 (1 = langsamste, 9 = schnellste Ausführung) eingeben. Nachdem man **G** eingegeben hat, ist die vorgewählte Geschwindigkeit 5. Die Geschwindigkeit kann auch schon vor Beginn der Ausführung gewählt werden. Die Programmausführung dauert solange an, bis sie auf die unten beschriebene Weise unterbrochen wird.

Z startet die Programmausführung ohne Animation. Sobald ein DISPLAY UPON CRT oder ACCEPT FROM CRT durchlaufen wird, wird der Quellcode durch die Bildschirmanzeige des Programms ersetzt und bleibt solange am Bildschirm, bis die Programmausführung gestoppt wird.

S hält die Programmausführung an und zeigt die Bildschirmausgabe des Benutzerprogramms an.

Wurde die Programmausführung mit einem der oben beschriebenen Kommandos gestartet, kann sie auf folgende Weise wieder angehalten werden:

- Wenn die Leertaste gedrückt wird, wird die Programmausführung sofort gestoppt.
- Wird ein vorher gesetzter Unterbrechungspunkt erreicht, wird die Programmausführung angehalten.
- Wird eine mit dem T-Kommando gestellte Bedingung wahr, wird die Programmausführung ebenfalls angehalten.
- Wird ein STOP RUN-Befehl erreicht, wird folgende Zeile ausgegeben:

WARNING - Next instruction is STOP RUN - S(top) C(ontinue)

Mit S können Sie das Programm beenden, mit C fortsetzen.

- Tritt ein Laufzeitfehler auf, wird folgende Zeile ausgegeben:

RTS ERROR: nnn - S(top) C(ontinue)

Mit S können Sie das Programm beenden, mit C fortsetzen.

Die Fehlernummern sind ausführlich in der LEVEL II COBOL Bedienungsanleitung beschrieben.

**L Level**

Normalerweise durchläuft die Programmanimation jede Schachtelungsstufe von PERFORM-Aufrufen. Mit diesem Kommando können Sie nun die Schachtelungstiefe begrenzen. Ineinandergeschachtelte PERFORMS, die über dieser angegebenen Grenze liegen, werden als ein einziges Statement angesehen, d.h. der Cursor wird bei der Animation nicht in diese von PERFORM aufgerufene Prozedur geführt.

Wenn Sie **L** eingegeben wird das folgende Untermenü angezeigt:

PERFORM LEVEL = , THRESHOLD LEVEL = S(et) U(nset) E(xit) Q(uit)

Es werden sowohl die derzeitige PERFORM-Schachtelungstiefe als auch die derzeitig gesetzte Grenze angezeigt.

Sie können folgende Kommandos eingeben:

- S** begrenzt die Schachtelungstiefe auf die Stufe, auf der der Cursor steht.
- U** nimmt die angegebene Begrenzung wieder zurück. Die Animation erfolgt wieder in allen Stufen.
- E** beendet die Programmausführung des aktuellen PERFORM ohne Animation, setzt den Cursor auf den Befehl, der dem PERFORM folgt und setzt die Begrenzung auf die Ebene dieses Befehls.
- Q** bewirkt einen sofortigen Abbruch der aktuellen PERFORM-Routine, ohne darin weitere Befehle auszuführen. Dieses Kommando erlaubt Ihnen auch ein gezieltes Verlassen einer Schleife in einer PERFORM-Routine.

### P Program Counter

Mit diesem Kommando können Sie sich darüber informieren, an welchem Punkt die Ausführung gestartet oder wieder aufgenommen wird, oder aber diesen Punkt verändern. Um den Startpunkt zu verändern, müssen Sie erst den Cursor an die entsprechende Stelle im Programm setzen.

Wenn Sie P eingeben, wird folgendes Untermenü angezeigt:

PROGRAM COUNTER - W(here) R(eset)

Folgende Kommandos stehen Ihnen dann zur Verfügung:

W verändert das Fenster entsprechend und positioniert den Cursor auf den nächsten auszuführenden Befehl. Diese Funktion ist dann nützlich, wenn Sie zwischenzeitlich das Fenster mit S verändert haben und nun wissen wollen, an welcher Stelle die Programmausführung weitergeht.

Sie ist aber nicht notwendig, da ANIMATOR die Ausführung des Programms immer an der richtigen Stelle wiederaufnimmt, außer das folgende Kommando R wird benutzt.

W kann auch aus dem Hauptmenü heraus direkt eingegeben werden (d.h. ohne vorher P einzugeben).

R An der aktuellen Cursorposition wird das Programm gestartet oder fortgesetzt. Vor der Eingabe von P muß der Cursor auf das erste Zeichen des ausführbaren Befehls (d.h. ein COBOL-Wort) gebracht werden.

**N Name**

Läuft der ANIMATOR ab, werden alle Programme der Ausführungseinheit, die mit ANIM übersetzt wurden, mit animiert.

Vielleicht wollen Sie nicht alle Programme mit ANIMATOR ablaufen lassen. Dann können Sie mit diesem Kommando genau bestimmen, welche Programme animiert werden.

Wenn Sie **N** eingeben, wird folgendes Untermenü angezeigt:

PROGRAM NAME - W(hich) A(ny) T(his) O(ther)

Folgende Kommandos stehen Ihnen zur Verfügung:

- W** zeigt den Namen des laufenden Programms an.
- A** bewirkt, daß alle für die Animation vorbereiteten Programme animiert werden (= Voreinstellung).
- T** bewirkt, daß nur das derzeit laufende Programm animiert wird. Alle anderen Programme laufen ohne Animation ab.
- O** Hiermit können Sie den Namen des Programms eingeben, das als nächstes animiert werden soll. Schließen Sie die Eingabe mit **↓** ab. Die Eingabe muß als ein alphanumerisches Literal eingegeben werden, d.h. sie muß in Anführungszeichen eingeschlossen werden. Die Animation des laufenden Programms wird bis zum Ende fortgesetzt. Alle anderen Programme, mit Ausnahme des angegebenen, werden ohne Animation ausgeführt.

**Anmerkung**

Wenn Sie wünschen, daß das Programm bis zu einem bestimmten Unterprogramm ohne Animation ablaufen soll, so erreichen Sie dies, indem Sie beim Aufruf des ANIMATOR folgende Option angeben:

```
animate programmname ZOOM \"(unterprogrammname)\"
```

### 3.5 Überprüfen und Ändern von Daten

- Die folgenden Kommandos geben Ihnen die Möglichkeit, den Inhalt bestimmter Datenfelder zu überprüfen und zu verändern. Sie wählen die gewünschten Datenfelder aus, indem Sie entweder den Namen des Datenfeldes eingeben oder den Cursor auf den Feldnamen im Programm positionieren.

#### D Display

- Mit diesem Kommando können Sie Datenfelder anzeigen und verändern, indem Sie in der Kommandozeile den Namen eingeben. Geben Sie **D** und anschließend den Namen des gewünschten Datenfeldes ein. Beenden Sie die Eingabe mit **↵**.


Der Wert des angegebenen Datenfeldes wird angezeigt, wobei eine der PICTURE-Klausel entsprechende Umformung stattfindet. Alphanumerische Datenfelder oder Datengruppen, die nicht dem ASCII-Code entsprechen, werden als Leerstelle angezeigt.

Für numerische Datenfelder werden, je nach Definition, Vorzeichen voran- oder nachgestellt. Dezimalpunkte werden, wenn nötig, eingefügt.

- Nach Ausgabe des Datenfeld-Inhalts wird der Cursor in die nächste Zeile positioniert, damit Sie ggf. einen anderen Wert für dieses Feld eingeben können. Wenn Sie keine Veränderung wünschen, dann drücken Sie einfach **↵**, andernfalls geben Sie den neuen Wert im COBOL-Literalformat ein (alphanumerische Literale müssen in Anführungszeichen eingeschlossen werden). Der Austausch des Wertes wird entsprechend den Regeln des MOVE-Befehls ausgeführt.



## Query

Mit diesem Kommando können Sie sich den Inhalt eines Feldes ausgeben lassen und es verändern. Bringen Sie den Cursor an irgendeine Stelle des Quellprogramms, an der der Feldname steht und geben Sie dann  ein.

Der Feldname wird angezeigt und kann, wie bei dem D-Kommando beschrieben, verändert werden.

### *Anmerkung*

Es können nur die ersten 80 Zeichen eines Datenfeldes ausgegeben werden.

## M Monitor

Mit diesem Kommando können Sie sich nach jedem Befehl den Inhalt eines bestimmten Feldes anzeigen lassen (ohne die Möglichkeit, ihn zu verändern).

Nach Eingabe von M wird folgendes Untermenü angezeigt:

MONITOR - S(et) U(nset) N(ame)

Sie haben folgende Möglichkeiten:

- S setzt die Monitorfunktion auf das Datenfeld, unter dem der Cursor blinkt. Der Cursor muß, bevor Sie M drücken, auf irgendeinen dieser Datenfeldnamen im Quellprogramm gesetzt sein.
- U beendet die Monitorfunktion wieder.
- N Nachdem Sie N gedrückt haben, können Sie den gewünschten Datenfeldnamen eingeben. Beenden Sie die Eingabe mit J.

Der Inhalt des überwachten Feldes wird nach jedem Animationsschritt, d.h. nach jedem Befehl im "GO"-Modus, angezeigt.



### 3.6 Bildschirmanzeige des Benutzerprogramms

Während der Animation werden alle Aus- bzw. Eingaben von DISPLAY- bzw. ACCEPT im ANSI-Format am unteren Bildschirmrand vorgenommen, um die Verwendung des Bildschirms durch den ANIMATOR nicht zu beeinträchtigen.

Dieses Vorgehen ist für die erweiterten Befehle ACCEPT FROM CRT und DISPLAY UPON CRT nicht möglich, da sie den ganzen Bildschirm benötigen. Aus diesem Grund werden die Bildschirmdaten dieser ACCEPT- bzw. DISPLAY-Befehle intern zwischengespeichert. Dieser Benutzerbildschirm überschreibt dann das Fenster zum Quellprogramm, wenn ein ACCEPT durchlaufen wird, so daß Sie ganz normal mit dem Bildschirm arbeiten können.

Darüberhinaus können Sie sich den Bildschirm des Programms anzeigen lassen, indem Sie  eingeben.

#### User

Wenn Sie  eingeben, wird der Benutzerbildschirm angezeigt. Das Fenster zum Quellprogramm wird dabei überschrieben. Der Benutzerbildschirm bleibt solange bestehen, bis Sie irgendeine Taste drücken, dann wird der Quellcode wieder angezeigt.

## 4 Die Kommandos in alphabetischer Reihenfolge

☾ Dies ist eine Zusammenfassung der Kommandos in alphabetischer Reihenfolge, die Sie aus der Hauptkommandozeile des ANIMATORS auswählen können. Bei jedem Kommando steht eine Kurzbeschreibung.

**A** (locAte)  
Mit diesem Kommando können Sie die Definition eines Datennamens bzw. eines Prozedurnamens im Quellprogramm finden, indem Sie den gesuchten Namen eingeben. Der Cursor wird auf die gesuchte Definition positioniert.

☾  **B** (Break-point)  
Mit diesem Kommando können Sie Unterbrechungspunkte im Programm definieren, an denen der Programmablauf anhalten soll. Zu diesem Kommando gibt es in einem Untermenü 4 Folgekommandos:

**S** (Set)  
Ein Unterbrechungspunkt wird an dem Programm-befehl gesetzt, an dem der Cursor gerade steht.

**U** (Unset)  
Zurücknehmen des Unterbrechungspunktes, auf dem der Cursor gerade steht.

**C** (Cancel)  
Zurücknehmen aller gesetzten Unterbrechungspunkte.

☾  **X** (eXamine)  
Anzeigen des nächsten Unterbrechungspunktes. Mit  können nacheinander alle gesetzten Unterbrechungspunkte abgefragt werden.

**I**  
Ein Unterbrechungspunkt mit einer Bedingung wird an der Stelle gesetzt, an der der Cursor gerade steht.

☾  **D** (Display)  
Mit diesem Kommando können Sie sich den Inhalt von Datenfeldern anzeigen lassen und ihn verändern.

## Kommandos, alphabetisch

---

- E** (Execute)  
Mit diesem Kommando wählen Sie aus, auf welche Weise das Programm ausgeführt werden soll. In einem Untermenü werden Ihnen 6 Möglichkeiten angeboten (mit Ausnahme der Auswahl S können diese Kommandos auch direkt aus dem Hauptmenü heraus aufgerufen werden):
- X** (eXecute)  
Ausführen eines Cobol-Befehls; der Cursor steht auf dem nächsten Befehl.
  - K** (sKip)  
Überspringen des nächsten Befehls ohne Ausführung. Der Cursor steht auf dem darauffolgenden Befehl.
  - I** (If)  
Ausführung des Programms ohne Animation bis zum nächsten If-Befehl. Das Programm hält an und der Cursor steht an diesem Befehl.
  - G** (Go)  
Starten der Programmausführung mit Animation. Sowie ein COBOL-Befehl ausgeführt ist, wird der Cursor im angezeigten Quellprogramm zum nächsten Befehl geführt. Die Ausführungsgeschwindigkeit kann bereits vor der Eingabe von **G** bestimmt werden oder nach der Eingabe von **G** verändert werden, indem Sie 1 (am langsamsten) bis 9 (am schnellsten) eingeben.
  - Z** (Zoom)  
Starten der Programmausführung ohne Animation.
  - S** (Stop)  
Anhalten der Programmausführung unter ANIMATOR und Anzeige des letzten Bildschirms des Programms.
- F** (Find)  
Mit diesem Kommando können Sie ab der aktuellen Cursorposition im angezeigten Quellprogramm nach einer beliebigen Zeichenfolge suchen.

**L**

(Level)

Mit diesem Kommando können Sie die Schachtelungstiefe bestimmen, bis zu der aufgerufene PERFORM-Routinen vom ANIMATOR am Bildschirm angezeigt werden. Routinen auf tieferer Stufe werden nicht am Bildschirm angezeigt. Der Cursor bleibt auf dem PERFORM-Befehl stehen.

Sie können folgende Kommandos eingeben:

**S**

(Set)

Begrenzen der Schachtelungstiefe auf den aktuellen Level.

**U**

(Unset)

Aufheben der Begrenzung, d.h. die Animation erfolgt wieder in allen Ebenen.

**E**

(End)

Fortsetzen der laufenden PERFORM-Routine ohne weitere Animation. Die Begrenzung der Schachtelungstiefe wird auf die Ebene des dem PERFORM folgenden Befehls gesetzt.

**Q**

(Quit)

Sofortiges Beenden einer PERFORM-Routine, ohne einen weiteren Befehl auszuführen

**M**

(Monitor)

Mit diesem Kommando können Sie sich während des Programmablaufs nach jedem Befehl den Inhalt eines Datenfeldes anzeigen lassen. In einem Untermenü werden Ihnen folgende Möglichkeiten geboten:

**S**

(Set)

Einschalten der Monitorfunktion für das Datenfeld, auf dem der Cursor gerade steht.

**U**

(Unset)

Ausschalten der Monitorfunktion

**N**

(Name)

Einschalten der Monitorfunktion durch Eingabe des Namens des Datenfeldes, das überwacht werden soll.

## Kommandos, alphabetisch

---

- N** (Name)  
Festlegen des (Unter-)Programms, das animiert werden soll
- O** (IOcate)  
Mit diesem Kommando können Sie die Definition eines Datennamens bzw. eines Prozedurnamens im Quellprogramm finden, indem Sie den Cursor auf den Namen des gewünschten Feldes irgendwo im angezeigten Quellprogramm setzen. Der Cursor wird auf die gesuchte Definition positioniert.
- P** (Program counter)  
Mit diesem Kommando können Sie feststellen, an welcher Stelle das Programm fortgesetzt wird, und diese Stelle verändern. Folgende weiteren Kommandos werden Ihnen in einem Untermenü geboten:
- W** (Where)  
Setzt den Cursor auf den nächsten auszuführenden Befehl.
- R** (Reset)  
Das Programm wird genau dort fortgesetzt, wo der Cursor gerade steht.
- Q** (Query)  
Dieses Kommando gibt Ihnen den Inhalt des Feldes aus, auf dessen Namen der Cursor vor Eingabe von **Q** gerade steht. Den Inhalt können Sie anschließend verändern.
- S** (Screen)  
Mit diesem Kommando können Sie das Fenster Ihres angezeigten Quellprogramms verschieben. Dabei gibt es folgende Kommandos:
- N** (Next)  
Nächste Seite des Quellprogramms anzeigen.
- P** (Previous)  
Vorhergehende Seite anzeigen.
- T** (Top)  
Anzeigen des Programmanfangs.
- E** (End)  
Anzeigen des Programmendes.

**V** (View)  
Verschieben des Quellprogramms, sodaß die Zeile, auf der der Cursor gerade steht, Zeile 3 wird.

*Anmerkung*

der Cursor muß vor Eingabe von **S** bereits positioniert sein.

**H** (Half)  
Unterteilen des Bildschirms in 2 Hälften. Das untere Fenster zeigt den Programmanfang.

*Anmerkung*

Hierauf folgende Bildschirmkommandos beziehen sich immer auf das Fenster, in dem der Cursor gerade steht.

**F** (Full)  
Zurück zu einem einzigen Fenster

**=n** Positionieren des Fensters so, daß die n-te Zeile des Quellprogramms die 3. Zeile des Fensters ist.

**+n** Verschieben des Fensters nach oben um n Zeilen

**-n** Verschieben des Fensters nach unten um n Zeilen

*Anmerkung*

nach Eingabe von **=** **+** **-** wird der Cursor in die Kommandozeile zur Eingabe von n positioniert. Beenden Sie die Eingabe mit **↓**.

Es gibt einen Spezialfall beim S-Kommando: Steht bei geteiltem Bildschirm der Cursor auf der Trennungslinie der beiden Fenster, wenn Sie **S** eingeben, so erscheint folgendes Untermenü:

SCREEN DIVIDER - U(p) D(own)

Sie können nun die Größe der beiden Fenster verändern, indem Sie die Trennungslinie verschieben mit

**U** 1 Zeile nach oben

**D** 1 Zeile nach unten.

## Kommandos, alphabetisch

---

**T**

(unTil)

Mit diesem Kommando können Sie eine COBOL-Bedingung definieren. Wenn diese Bedingung erfüllt ist, wird die Ausführung des Programms angehalten. Sie können folgende Kommandos eingeben:

**S**

(Set)

Eingabe der gewünschten Bedingung im COBOL-Format, gefolgt von **↓**.

**U**

(Unset)

Löschen einer vorher eingegebenen Bedingung.

**X**

(eXamine)

Anzeigen einer vorher gesetzten Bedingung.

**U**

(User)

Mit diesem Kommando können Sie sich den augenblicklichen Benutzerbildschirm des Programms anzeigen lassen. Sobald Sie irgendeine Taste drücken, wird wieder das Quellprogramm angezeigt.

## 5 Kommandos, kurz und bündig

- ☐ A Datendefinition suchen über den Namen
- ☐ B Unterbrechungspunkt setzen
  - ☐ S Cursorstelle ist Unterbrechungspunkt
  - ☐ U Hebt den Unterbrechungspunkt auf
  - ☐ C Löscht alle Unterbrechungspunkte
  - ☐ X Zeigt alle Unterbrechungspunkte an
  - ☐ I Unterbrechungspunkt mit Bedingung
- ☐ D Datenfelder anschauen und verändern
- ☐ E ANIMATOR laufen lassen
  - ☐ X Pro Eingabe 1 Befehl weiter
  - ☐ K 1 Befehl überspringen
  - ☐ I Ohne Animation bis zum nächsten IF
  - ☐ G Ablauf mit Geschwindigkeit 1 - 9
  - ☐ Z Erst Halt bei DISPLAY UPON CRT oder ACCEPT FROM CRT
  - ☐ S Stop und Bildschirmausgabe
 STOP RUN: ☐ S Ende ☐ C Weiter
- ☐ F Zeichenfolge suchen
- ☐ L Schachtelungstiefe bei Perform begrenzen
  - ☐ S Schachtelungstiefe begrenzen
  - ☐ U Begrenzung zurücknehmen
  - ☐ E Ohne Animation auf Befehl nach PERFORM und Begrenzung auf diese Ebene setzen
- ☐ Q Ausprung aus dem aktuellen PERFORM
- ☐ M Monitorfunktion einschalten
  - ☐ S Überprüfung des markierten Feldes ein
  - ☐ U Überprüfung ausschalten
  - ☐ N Überprüfung des genannten Feldes ein
- ☐ N Programme zur Animation auswählen



## kurz und bündig

---

- O Datendefinition suchen über die Cursorposition
- P Programmzähler erfragen oder setzen
- W Fenster auf nächstem zu durchlaufenden Befehl  
 R Wo Cursor steht, geht's weiter
- Q Datenfeld abfragen und setzen
- S Fenster verändern
- N Nächste Seite im Quellprogramm
- P Vorherige Seite im Quellprogramm
- T Anfang des Quellprogramms
- E Ende des Quellprogramms
- V Cursorzeile ist dann 3. Zeile
- H 2 Fenster
- F Wieder 1 Fenster
- =  n n (= Zeilennummerierung) ist 3. Zeile
- +  n um n Zeilen weiter vorwärts
- n um n Zeilen weiter rückwärts
- S Fenstertrennlinie verändern
- U 1 Zeile nach oben
- D 1 Zeile nach unten
- T Unterbrechungsbedingung definieren
- U Benutzerbildschirm anzeigen
- W Namen des laufenden Programms anzeigen
- A Alle Programme werden animiert
- T NUR das aktuelle Programm wird animiert
- O Namen vom nächsten Programm eingeben, das animiert werden soll
- S Bedingung eingeben
- U Bedingung löschen
- X Bedingung anzeigen
- ? Anderer Teil des Hauptmenüs