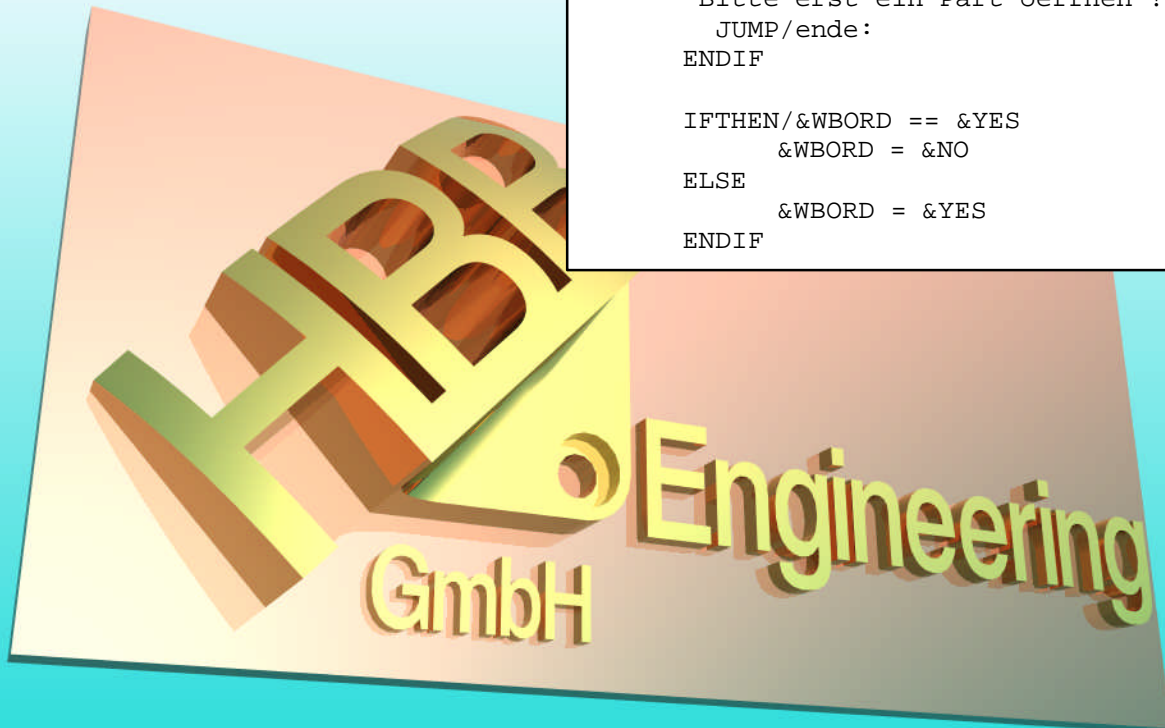


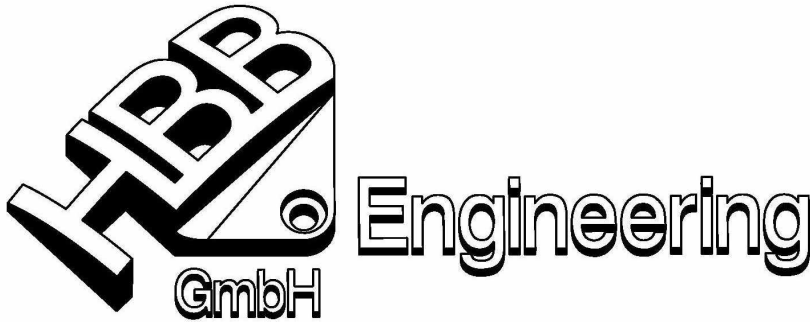
# UNIGRAPHICS

## GRIP - Handbuch

```
$$  
$$ Programmierer:      Walter Hogger  
$$  
$$ Erstelldatum:      20.10.1998  
$$
```

```
$$--- Ueberpruefung Aktives Part  
      IFTHEN/&ACTPRT == 1  
          MESSG/'Das Programm benoetigt  
ein aktives Part.', $  
          'Bitte erst ein Part oeffnen !'  
          JUMP/ende:  
      ENDIF  
  
      IFTHEN/&WBORD == &YES  
          &WBORD = &NO  
      ELSE  
          &WBORD = &YES  
      ENDIF
```





*HBB Engineering GmbH  
Salzstraße 9  
D-83454 Anger*

*Telefon 08656-98488-0  
Telefax 08656-98488-88  
Info@HBB-Engineering.de  
www.HBB-Engineering.de*

Stand: 03/2004

# UNIGRAPHICS

## GRIP - Handbuch

[HBB\_GRIP\_Handbuch.doc]

**GRIP** steht für **GR**aphics **I**nteractive **P**rogramming.

Die Sprache wurde generiert, um Ingenieuren und Konstrukteuren die Möglichkeit zu bieten, spezielle Abläufe, Funktionen und Vorgehensweisen in Programmform zu bringen. Die Sprache wurde nicht für professionelle Programmierer entwickelt, ist also eher „primitiv“ gehalten (aus der Sicht von professionellen Programmierern).

Wer GRIP-Programme entwickeln will, muß UNIGRAPHICS kennen! GRIP bietet sehr viele Sprachworte, welche Aktionen auslösen oder Objekte erzeugen, die der „UNIGRAPHICS-Normalbenutzer“ auch interaktiv erzeugen könnte. Wer nicht weiß, was UNIGRAPHICS für Möglichkeiten bietet, tut sich auch schwer, in GRIP die entsprechenden Programmwünsche umzusetzen. Daraus leitet sich die **Empfehlung** ab, daß vor der Erstellung eines Programms **erst in UNIGRAPHICS die Funktion interaktiv durchforstet** werden sollte, da sehr viele – nicht alle! - GRIP-Befehle genauso aufgebaut sind wie die interaktiven Befehle (gleiche Optionen, gleicher Variationen, etc.).

Nicht für alle interaktiven Funktionen von UNIGRAPHICS gibt es entsprechende GRIP-Kommandos. Umgekehrt kann man in GRIP einige Aktionen durchführen, an die der normale UNIGRAPHICS-Benutzer nicht herankommt.

GRIP-Programme müssen kompiliert und gelinkt (gebunden) werden, um für den Anwender aufrufbar zu werden.

Zum Ablauf der Programme ist die „GRIP-Runtime-Licence“ nötig. Zum Erstellen der Programme ist noch zusätzlich die „GRIP-Development-License“ nötig.

Es ist dringend zu empfehlen, während der Programmentwicklung die UNIGRAPHICS-Online-Hilfe zur Verfügung zu haben, da hier die Syntax der einzelnen Kommandos nachgeschlagen werden kann.

GRIP-Programme waren in den letzten 20 Jahren sehr gut aufwärtskompatibel. Es laufen in UG NX2 noch Programme, die vor 20 Jahren mit UG V2 entwickelt wurden. Ggf. müssen die Programme neu kompiliert und gelinkt werden.

Die GRX-Dateien müssen im allgemeinen auf jeder Rechnerplattform (HP-UX, AIX, UNIX, Windows NT, ...) neu erzeugt werden. Es laufen jedoch auch Programme die z.B. auf einer HP-Maschine (HP-UX, UNIX) erzeugt wurden auch auf SGI (Silicon Graphics).

Die 2. Programmiersprache zu UNIGRAPHICS nennt sich „User Function“ (UFUNC). Dabei handelt es sich um reine C-Programme. Als Compiler dient ein „normaler“ ANSI-fähiger Compiler (z.B. von Microsoft). Hier werden dem Programmierer zahlreiche Funktionen angeboten, welche UNIGRAPHICS-Funktionen darstellen. UFUNC geht weiter als GRIP, ist jedoch aufwendiger zu programmieren. In UFUNC stehen dem Programmierer nahezu dieselben Funktionen zur Verfügung, wie den Entwicklern von UNIGRAPHICS. Die Möglichkeiten von User Function sind umfangreicher wie die von GRIP. Beispielsweise ist es in GRIP nicht möglich, die „Part History“ eines Teiles zu löschen, in UFUNC gibt es eine Funktion dafür. Zahlreiche andere Beispiele könnten hier folgen.

Der typische Weg beim Erstellen von GRIP-Programmen ist folgender:

- 1) Programm mit einem ASCII-Editor schreiben (Ergebnis: \*.grs, GRIP Source)
- 2) Programm übersetzen/kompilieren (Ergebnis: \*.gri, GRIP Intermediate)
- 3) Programm binden/linken (nur das Hauptprog.) (Ergebnis: \*.grx, GRIP Execute)

Nur die „GRX-Datei“ ist für den Einsatz nötig (ggf. jedoch auch Partfiles oder Tabellen, aus denen Werte entnommen werden).

### **Generelles Format eines GRIP-Statements:**

Major word / parameters

z.B. DELETE / ln1

Variable = Major word / parameters

z.B. p1 = POINT / 0,10,4.6

Variable1 = Major word / Variable2, Minor word, Variable3

z.B. l1 = LINE / p1, LEFT, p2, TANTO, c1

**Deklarationen:**

ENTITY / p1, L1, LINIE2, punkt(10)	grafische UG-Elemente, z.B. Punkte, Linien, Flächen, Körper, Bemaßungen, Text, ...
STRING / zeile(80), str(2), feld1(80,100)	Textfelder
NUMBER / x1, ykoord(10), ss(10,10)	Werte, Zahlen, Felder sind zwingend anzugeben, Einzelwerte optional
DATA / msg1, 'Zeige Punkt', \$ msg2, 'welche Farbe ?'	Wertzuweisung für Werte und Texte, „\$“ = Folgezeile

**Variablenamen:**

- maximal 32 Zeichen
- erstes Zeichen muß ein Buchstabe sein
- kein Haupt- oder Nebenwort (Major- / Minor word) verwenden
- nur die ersten sechs Zeichen sind relevant (im Compiler-Listing)
- Klein- / Großschreibung ist egal
- Unterstrich „\_“ ist erlaubt

z.B:	p1	ok
	Linie_1	ok
	LL1(4)	ok, einzelnes Feldelement
	Koordinate_Startpunkt	ok
	1p	falsch, 1. Zeichen kein Buchstabe
	LINE	falsch, GRIP-Sprachwort
	Entity_Zaehler	falsch, 1. sechs Zeichen ein GRIP-Sprachwort
	Dies_ist_auch_falsch_da_mehr_als_32_Zeichen	

**Sonstige Syntax:**

**Leerzeilen** können beliebig viele verwendet werden.

**Kommentare** beginne mit „\$\$“ (auch innerhalb einer Programmzeile möglich).

Bei **Folgezeilen** ist das letzte Zeichen ein „\$“.

Innerhalb **von Folgezeilen** sind **keine Kommentarzeilen** erlaubt.

**Haupt- und Unterprogramme:****Hauptprogramm**

```

ENTITY/kreis
STRING/name1(30)
...
$$--- Unterprogrammaufruf mit Uebergabeparameter
CALL/`kreis_check`,kreis,name1,retcod

IF retcode <> 0, JUMP/err007:

...
HALT

```

**Unterprogramm „kreis\_check.grs“**

```

PROC/kreis,name1,retcod

ENTITY/kreis          $$ Dimensionierung im Hauptprogramm
STRING/name1          $$ Dimensionierung im Hauptprogramm

...
IF/LENF(name1) < 1, retcod = -1
...
RETURN

```

Die Variablen „kreis“ und „name1“ sind Eingabeparameter für das Unterprogramm „kreis\_check“, die Variable „retcod“ ist ein Ausgabeparameter (Rückgabewert).

Die Dimensionierung erfolgt im Hauptprogramm.

Die Reihenfolge der Parameter muß im Unterprogrammaufruf und in der „PROC/-Zeile“ identisch sein, die Variablenname könnten voneinander abweichen. **Nur die Reihenfolge zählt, nicht der Name!**

Alle Programme müssen kompiliert werden.

Nur das Hauptprogramm wird gelinkt.

**Programmbeispiele für die interaktiven Kommandos:**

```

$$-----
$$
$$ GRIP - Kurs    / Walter Hogger /    8/96
$$
$$-----
$$
$$ choose_demo.grs
$$
$$ Beispiel des GRIP-Sprachwortes "CHOOSE".
$$
$$ W. Hogger / 28.08.1996
$$

```

```

start:

```

```

    CHOOSE/'Waehle einen Wochentag !' ,      $
        'Montag',                          $
        'Dienstag',                        $
        'Mittwoch',                       $
        'Donnerstag',                     $
        'Freitag',                       $
        'Samstag',                       $
        'Sonntag',                       $
        '',                              $
        'ENDE',                          $
        DEFLT,1,antwort
    JUMP/ende:,ende:,,antwort

```

```

antwort = antwort - 4

```

```

    IFTHEN/ antwort == 1
        MESSG/'Die Woche ist noch lang ...'
    ELSEIF/ antwort == 2
        MESSG/'Lottospielen nicht vergessen !'
    ELSEIF/ antwort == 3
        MESSG/'Heute waer ein Feiertag recht ...'
    ELSEIF/ antwort == 4
        MESSG/'Nicht mehr lang ...'
    ELSEIF/ antwort == 5
        MESSG/'Endlich Freitag !'
    ELSEIF/ antwort == 6
        MESSG/'Was machste heute in der Fiiierma ?'
    ELSEIF/ antwort == 7
        MESSG/'Liest sowieso keiner am Sonntag !'
    ELSE
        MESSG/'Programmende'
    JUMP/ende:
ENDIF

```

```

$$--- nochmal ...
    JUMP/start:

```

```

ende:    HALT

```

[illegible]

```

$$-----
$$
$$ GRIP - Kurs    / Walter Hogger /    9/96
$$
$$-----
$$
$$ mchoose_demo.grs
$$
$$ Beispiel des GRIP-Sprachwortes "MCHOOS".
$$
$$ W. Hogger / 25.09.1996
$$

```

```

        STRING/Cuetext(40)      $$ Aufforderungstext
        STRING/Angebot(7,40)    $$ Optionen
        STRING/Alternat(40)     $$ Alternate-Action-Text
        NUMBER/Antwort(7)       $$ ARRAY fuer die
                                $$ moeglichen Antworten

```

```

$$---  Vorbelegungen
        Cuetext      = 'Hier kann man knipsen ...'
        Angebot(1)   = 'Montag'
        Angebot(2)   = 'Dienstag'
        Angebot(3)   = 'Mittwoch'
        Angebot(4)   = 'Donnerstag'
        Angebot(5)   = 'Freitag'
        Angebot(6)   = 'Samstag'
        Angebot(7)   = 'Sonntag'
        Alternat     = 'Alle Knoepfe wieder raus.'

```

```

$$$      Antwort(1) = 1  $$ klappt nicht! d.h. die Knopfstellung
          $$ kann nicht vorbelegt werden

```

```

start:
        MCHOOS/Cuetext,Angebot,Antwort,ALTACT,Alternat,antw1
        JUMP/start:,ende:,,antw1

```

```

$$--- Auswerten der Antworten
$$---  0 = Knopf wurde nicht gedrueckt
$$---  1 = Knopf wurde gedrueckt
        DO/loop1:,i,1,7
          IF/ Antwort(i) == 1,      $
            PRINT/'gewaehlt wurde -> ' + Angebot(i)
loop1:

```

```

$$--- nochmal ...
        JUMP/start:

```

```

ende:    HALT

```



```

$$-----+
$$
$$ GRIP - Kurs      / Walter Hogger /      8/96
$$
$$$$
$$$$
$$$$ GRIP-Programm: crt wrt_demo.grs
$$$$
$$$$-----
$$$$
$$$$ Das Programm dient als Beispiel fuer die Verwendung
$$$$ der GRIP-Sprachworte "CRTWRT/" und "POS/".
$$$$
$$$$ Programmierer: W. Hogger
$$$$
$$$$ Datum:          29.08.1996
$$$$
$$$$ Aenderung:      -
$$$$
$$$$
$$$$
$$$$--- Deklarationen
        NUMBER/xyz(3)           $$ Feld fuer x-y-z-Koordinaten
        STRING/meldung(40)       $$ Text max. 40. Zeichen

start:

$$$$--- Benutzer soll Text eingeben
        TEXT/'Welche Meldung willst du sehen?',meldung,antw
        JUMP/ ende:, ende: , , antw
$$$$_____ back __ cancel __ ok __ alternative

$$$$--- Falls nichts eingegeben wurde ...
        IFTHEN/ LENF(meldung) < 1
            MESSG/'Ok, wenn Du nicht willst ...',$
                'Programmabbruch'
        JUMP/ende:
    ENDIF

$$$$--- Benutzer soll die Stelle der Meldung zeigen
$$$$--- "POS" -> Bildschirmposition, "GPOS" -> Punktemenue
        POS/'Wo soll die Meldung erscheinen?', xyz, antw
        JUMP/ start:, ende: , , antw
$$$$_____ back __ cancel __ ok __ alternative

$$$$--- Meldung anschreiben
        CRTWRT/meldung,xyz

$$$$--- Programm noch einmal durchlaufen
        JUMP/start:

ende: HALT

```

**Auszüge aus der Online – Hilfe: Major Words**

<u>Major Word</u>	<u>Function Description</u>	<u>Category</u>
<u>ABSF</u>	Absolute Value	System/Calculator
<u>ACOSF</u>	Angle Whose Cosine Is arg	System/Calculator
<u>ADDPL</u>	Add Row	System/Assemblies
<u>ADIM</u>	Angular	Dimensioning/Drafting Objects
<u>ANGLF</u>	Angle of a Line	System/Verify Properties
<u>ANLSIS</u>	Rotate about XC or YC Axis	System/3D Analysis
<u>ANLSIS</u>	Project along ZC Axis	System/3D Analysis
<u>ANLSIS</u>	Bounded by Sheets	System/3D Analysis
<u>ANLSIS</u>	Thin Shell	System/3D Analysis
<u>ANLSIS</u>	Mass Properties of Solid Bodies	System/3D Analysis
<u>ANLSIS</u>	Arc Length	System/3D Analysis
<u>ANLSIS</u>	2D Analysis	System/Verify Properties
<u>APPEND</u>	File Pointer Control	File/Edit
<u>ARCDIM</u>	Arc Length	Dimensioning/Drafting Objects
<u>ASATT</u>	Assign Object Attributes	System/Attributes
<u>ASCENT</u>	Associated Object of Dim/Draft Object	System/3D Analysis
<u>ASCII</u>	ASCII Value of a Character	Strings
<u>ASGNFT</u>	Assign Font	System/Parameters
<u>ASINF</u>	Angle Whose Sine Is arg	System/Calculator
<u>ATANF</u>	Angle Whose Tangent Is arg	System/Calculator
<u>AUTOSE</u>	Create Sheets from Curves	Sheet Body/Body Surface

**Auszüge aus der Online – Hilfe: Minor Words**

*Minor Word List*

ACCRCY  
ACTIVE  
ADD  
AFILL  
AFTER  
AITPZ  
ALL  
ALTACT  
ALWAYS  
AMBI  
ANCHOR  
ANGLE  
ANTALI  
APEX  
APPEND  
APPROX  
ARCLEN  
ARROW  
ARROWS  
ASCEND  
ATANGL  
ATRBTS  
AUTO  
AXIS  
BACK  
BACKZ  
BEFORE  
BLANK  
BLEND  
BMDRY  
BND  
BOTH

**Auszüge aus der Online – Hilfe: GPA Symbols**

<u>GPA Symbol</u>	<u>Function Description</u>	<u>Category</u>
<u>&amp;ABS</u>	Absolute Coordinate System	Geometry
<u>&amp;ACTPRT</u>	Active Part Status	File Management
<u>&amp;ALARND</u>	All Around Symbol Display	Drafting
<u>&amp;ANGTOL</u>	Modeling Angle Tolerance	Geometry
<u>&amp;APSITE</u>	Appended Text Site	Dimension
<u>&amp;ARCCTR</u>	Radius Dimension Display	Dimension
<u>&amp;ARDSP</u>	Arrow Display	Dimension
<u>&amp;ARFLDT</u>	Filled Arrowhead Distance	Dimension
<u>&amp;ARIANG</u>	Arrowhead Included Angle	Dimension
<u>&amp;ARLNSM</u>	Arc Length Symbol	Dimension
<u>&amp;ARROW</u>	Arrowhead 1 Type	Dimension
<u>&amp;ARROW2</u>	Arrowhead 2 Type	Dimension
<u>&amp;ASIZE</u>	Arrowhead Size	Dimension
<u>&amp;ASPECT</u>	Aspect Ratio	Drafting
<u>&amp;ATTDIS</u>	Attribute Name Display	Attributes
<u>&amp;ATTTYPE</u>	Attribute Type	Attributes
<u>&amp;AUNT</u>	Angular Dimension Units	Dimension
<u>&amp;BATCH</u>	Current Execution Mode	System
<u>&amp;BGCLR</u>	Background Color	System
<u>&amp;CAREA</u>	Customer Area of the File Header	File Management
<u>&amp;CFONT</u>	Character Font	Dimension
<u>&amp;CHAIN</u>	Chaining Method	Chain
<u>&amp;CHTOL</u>	Chaining Tolerance	Chain

**Auszüge aus der Online – Hilfe: EDA Symbols**

<u>EDA Symbol</u>	<u>Function Description</u>	<u>Category</u>
<u>&amp;ALLARN</u>	All Around Symbol Display	Drafting & Dimensioning
<u>&amp;ALNSYM</u>	Arc Length Symbol Display	Drafting & Dimensioning
<u>&amp;ARWDAT</u>	Arrowhead Display Data	Drafting & Dimensioning
<u>&amp;ASCCNT</u>	Number of Associated Objects	Drafting & Dimensioning
<u>&amp;ATDISL</u>	Name Display Location (X,Y,Z)	Attributes
<u>&amp;ATTALL</u>	All Object Attributes	Attributes
<u>&amp;ATTTL</u>	Attribute Title	Attributes
<u>&amp;ATTVL</u>	Attribute Value	Attributes
<u>&amp;BLANK</u>	Blank	Geometry/Spline
<u>&amp;CCOUNT</u>	Number of Component Members	Geometry/Components & Reference Sets
<u>&amp;CDATE</u>	Component Date	Geometry/Components & Reference Sets
<u>&amp;CHRDAT</u>	Character Parameters	Drafting & Dimensioning
<u>&amp;CHRSET</u>	Native Language Menu Character Sets	Geometry/Spline
<u>&amp;CLRNME</u>	Color Name	Geometry/Spline
<u>&amp;CLRNUM</u>	Color Number	Geometry/Spline
<u>&amp;CMEM</u>	Object Identifier of Nth Member of Component	Geometry/Components & Reference Sets
<u>&amp;COLOR</u>	Color	Geometry/Spline
<u>&amp;CPNAM</u>	Component Part Name	Geometry/Components & Reference Sets
<u>&amp;CRSNAM</u>	Reference Set Name	Geometry/Components & Reference Sets
<u>&amp;CSCALE</u>	Component Scale	Geometry/Components & Reference Sets