

## **Dokumentation ueber die Befehlsebene (CLI)**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i>		
	Dokumentation ueber die Befehlsebene (CLI)		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 2, 2022	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Dokumentation ueber die Befehlsebene (CLI)</b>	<b>1</b>
1.1	CLI	1
1.2	CLI / Befehlsebene - Was ist das? - Wozu?	2
1.3	CLI / Eingabe von Befehlen	2
1.4	CLI / Besondere Zeichen in der Befehlszeile	3
1.5	CLI / Aufbau der Befehlszeile	4
1.6	CLI / Abkuerzungen, Gross/Kleinschreibung	6
1.7	CLI / Fehlermeldungen	6
1.8	CLI / Hilfstexte	7
1.9	CLI / Parametertypen	7
1.10	CLI / Symbole (Variablen), Abkuerzungen	9
1.11	CLI / Integer-Ausdruecke	10
1.12	CLI / Variablensubstitution	11
1.13	CLI / Zugriffsrechte Uebersicht	12
1.14	CLI / Zugriffsrechte durch Level	12
1.15	CLI / Zugriffsrechte durch Zugriffsflags (Acc)	13
1.16	Zugriffsvergabe fuer Optionen eines Befehles	13
1.17	Zugriffsrechte in Befehls-Batchdateien	14
1.18	CLI / Liste der Fehlermeldungen	15

---

## Chapter 1

# Dokumentation ueber die Befehlsebene (CLI)

## 1.1 CLI

Command Line Interface - Befehlsebene

-----  
Befehlsebene - Was ist das? - Wozu?

Grundlagen

Eingabe von Befehlen

Besondere Zeichen in der Befehlszeile

Aufbau der Befehlszeile

Abkuerzungen, Gross/Kleinschreibung

Fehlermeldungen

Hilfstexte

Parameter und Variablen

Parametertypen

Symbole (Variablen) und Abkuerzungen

Integer-Ausdruecke

Variablensubstitution

Zugriffsrechte

Zugriffsrechte Uebersicht

Zugriffsrechte durch Level

Zugriffsrechte durch Zugriffsflags

Zugriffsvergabe fuer Befehlsoptionen

Zugriffe in Befehls-Batchdateien

---

## Anhang

Liste der Fehlermeldungen  
Alle Rechte der Software AMMS gehoeren der Firma Marquardt & ←  
Scheler GbR.

## 1.2 CLI / Befehlsebene - Was ist das? - Wozu?

Befehlsebene - Was ist das ? - Wozu ?  
-----

Die Steuerung einer Mailbox ueber Menues ist zwar recht einfach und leicht zu erlernen, sie hat jedoch den Nachteil, dass man sich oft erst durch mehrere Menues hangeln muss, bis man die gewuenschte Funktion gefunden hat. Desweiteren ist es quasi unmoeglich, alle Funktionen eines Mailbox-programmes in Menues unterzubringen. Aus diesen Gruenden gibt es eine andere, schnelle und effektive, Moeglichkeit der Boxsteuerung: Befehle.

Ein kleines Beispiel: Angenommen, Sie wollen Ihre private Post lesen, so koennen Sie das auf zwei Arten tun:

- Im Menue-System, ausgehend vom Standart-Hauptmenue, waehlen Sie "P" fuer private Post und danach "R" fuer lesen. Das ganze geht auf der Console noch recht schnell, fuer einen User kann der Aufbau der Menues jedoch ziemlich langsam und nervend sein.
- In der Befehlsebene koennen Sie direkt den Befehl "MAIL" eingeben und landen prompt bei der privaten Post.

Es gibt eine Vielzahl von Befehlen und Funktionen, die ueber die Menues nicht ansprechbar sind, so dass das Arbeiten in der Befehlsebene fuer den User empfehlenswert und fuer den Sysop unerlaesslich ist. Aus diesem Grund moechte ich an dieser Stelle alle diejenigen, die sich bisher davor gescheut haben, ermutigen sich zumindest mit den Grundlagen der Befehlssteuerung vertraut zu machen.

Im Folgenden werden alle Befehlseingaben durch ein "AMMS> " als solche gekennzeichnet, Ausgaben von AMMS mit einem "->".

## 1.3 CLI / Eingabe von Befehlen

Eingabe von Befehlen  
-----

Dass Sie sich in der Befehlsebene von AMMS befinden, erkennen Sie an der Eingabeaufforderung am Anfang der Zeile, auch Prompt genannt. Immer wenn dieser Prompt erscheint, wartet AMMS auf die Eingabe eines Befehles. Ein Befehl kann aus mehreren Woertern, dem Befehlsnamen selbst und sogenannten Parametern (dazu spaeter mehr), bestehen. Im Folgenden nennen wir das ganze einfach Befehlszeile. Die Eingabe einer Befehlszeile wird durch die Return-Taste abgeschlossen. Dann ueberprueft AMMS die Zeile auf Richtigkeit und fuehrt entweder den entsprechenden Befehl aus oder gibt eine

Fehlermeldung aus wenn die Befehlszeile fehlerhaft war. Einige Tasten haben bei der Eingabe der Befehlszeile spezielle Funktionen zum Editieren:

TASTE	FUNKTION
<Return>	Eingabe abschliessen, Befehlszeile bearbeiten
<Ctrl-A>	Cursor an den Zeilenanfang positionieren
<Ctrl-E>	Cursor an das Zeilenende positionieren
<Ctrl-Z>	Befehlszeile loeschen
<Cursor links>	Cursor ein Zeichen nach links bewegen
<Cursor rechts>	Cursor ein Zeichen nach rechts bewegen
 <Cursor hoch> & <Cursor runter> :	

Mit diesen beiden Tasten koennen Sie sich eine der letzten 30 Befehlszeilen in die aktuelle Eingabezeile holen, um z.B. bei einer falschen Eingabe diese zu korrigieren und dann nochmal abzuschicken.

## 1.4 CLI / Besondere Zeichen in der Befehlszeile

### Besondere Zeichen in der Befehlszeile

-----

Innerhalb der Befehlszeile haben einige Zeichen eine besondere Bedeutung:

ZEICHEN	FUNKTION
;	Alles was hinter einem ';' in der Befehlszeile steht, wird ignoriert. Der Sinn dieses Zeichens liegt im Kommentieren von Befehlszeilen in Batchdateien.

Beispiel:

```
AMMS> echo "Hallo" ; Dieser Kommentar wird ignoriert
-> Hallo
```

-	Diese Befehlszeile in der naechsten Eingabezeile fortsetzen. Wozu dieses Zeichen gut ist, wird klar, wenn Sie eine Befehlszeile eingeben wollen, die laenger ist als die Eingabezeile. In diesem Fall geben Sie als letztes Zeichen ein '-' ein und schliessen die Eingabe mit Return ab, nun koennen Sie die selbe Befehlszeile in der naechsten Eingabezeile fortsetzen - AMMS fuegt danach beide Eingabezeilen zu einer Befehlszeile zusammen.
---	---

Beispiel: Aus der Eingabe

```
Prompt> list -
Cont.: user
```

fuegt AMMS die Befehlszeile "list user" zusammen und fuehrt diesen Befehl aus.

'	Dieses Zeichen dient zum Einsetzen von Variablen in die
---	---

---

Befehlszeile. Weiteres dazu siehe in dem Kapitel  
 Variablensubstitution  
 .

## 1.5 CLI / Aufbau der Befehlszeile

### Aufbau der Befehlszeile -----

Eine Befehlszeile hat folgenden grundsuetzlichen Aufbau:

```
<BEFEHL>[/OPTION] [PARAMETER] [PARAMETER] ...
```

Teile die in <> stehen, MUESSEN angegeben werden, Teile die in [] stehen sind optional, d.h. KOENNEN angegeben werden, um etwa die Funktion eines Befehles noch genauer zu spezifizieren. Wuerde man die Komponenten in "W-Formen" uebersetzen, so kann man sich die Bedeutung der einzelnen Teile in etwa so vorstellen :

```
<BEFEHL>   WAS soll gemacht werden
[/OPTION]  WIE soll es gemacht werden
[PARAMETER] WOMIT soll es gemacht werden bzw. WORAUF soll der Befehl
            angewendet werden.
```

Die Teile der Befehlszeile genauer betrachtet:

```
<BEFEHL>   Der eigentliche Befehl, er besteht entweder aus einem Wort,
            z.B. "DIR", oder aus zwei Woertern, z.B. "LIST USER".

[/OPTION]  Direkt hinter dem Befehl koennen, durch '/' getrennt, mehrere
            Optionen angegeben werden, um die Funktionsweise eines Befehls
            zu beeinflussen. Einige Beispiele:
```

Eingabe zur Ausgabe des Protokolles in der kurzen Form:

```
AMMS> PROTOCOL
      ^
      |_____ Befehlsname
```

Eingabe zur Ausgabe des Protokolls in der langen Form:

```
AMMS> PROTOCOL/FULL
      ^
      |_____ Option FULL
```

Es koennen auch mehrere Optionen hintereinander angegeben werden (soweit sinnvoll). Beispiel:

Eingabe zum Listen aller Directory-Inhalte mit Unterdirs:

```
AMMS> DIR/ALL
      ^
      |_____ Option ALL
```

Eingabe zum Listen aller Directory-Inhalte mit Unterdirs,  
jedoch OHNE Filebretter:

```
AMMS> DIR/ALL/NOFBOARDS
      ^   ^
      |   |_____ Option NOFBOARDS +
      |_____ Option ALL
```

Bei manchen Befehlen lassen sich durch die Optionen noch  
weitere Parameter angeben.

Beispiel: LIST USER-Befehl

Eingabe zum Listen aller User mit Level 9000:

```
AMMS> LIST USER/LEVEL=9000
      ^   ^   ^
      |   |   |_____ Option LEVEL mit Wertzuwei-
      |   |   |_____ sung.
      |   |_____ Befehlsname, zweites Wort
      |_____ Befehlsname, erstes Wort
```

[PARAMETER] Die Parameter geben an, worauf der Befehl angewendet werden  
soll. Welche Parameter angegeben werden MUESSEN und welche  
optional sind, haengt jeweils vom entsprechenden Befehl ab.

Beispiel: LIST USER-Befehl, Username OPTIONAL

Ohne Parameter werden ALLE User gelistet, Eingabe:

```
AMMS> LIST USER
```

OPTIONAL kann jedoch als Parameter ein Muster fuer die User-  
namen angegeben werden, die gelistet werden sollen.

Eingabe um alle User zu listen, deren Name mit A anfaengt:

```
AMMS> LIST USER a*
      ^
      |_____ Parameter (Username)
```

Beispiel: DELETE USER-Befehl, Username NICHT OPTIONAL

Bei diesem Befehl MUSS ein Username angegeben werden, da  
der Befehl ansonsten ja keinen Sinn macht. Um also z.B.  
den User TEST zu loeschen, waere die Eingabe:

```
AMMS> DELETE USER TEST
```

Wird ein solcher Parameter nicht angegeben, fragt AMMS danach:

```
AMMS> DELETE USER          <-- Parameter USERNAME fehlt
Gib Usernamen ein :       <-- AMMS fragt danach
```

Geben Sie in diesem Fall nichts ein (nur Return), wird der  
Befehl nicht ausgefuehrt und AMMS kehrt zur Befehlseingabe



zurueck.

## 1.6 CLI / Abkuerzungen, Gross/Kleinschreibung

### Abkuerzungen, Gross/Kleinschreibung

Bei der Eingabe von Befehlszeilen wird grundsaeztlich NICHT zwischen Gross- und Kleinschreibung unterschieden, eine Ausnahme ist die Zuweisung von Variablen (siehe bei  
 Symbole  
 ).

Befehls- und Optionsnamen lassen sich grundsaeztlich abkuerzen. Wieviele Buchstaben mindestens ausgeschrieben werden muessen, laesst sich bei der Ausgabe des ?-Befehls bereits ersehen: Die Teile vom Befehlsnamen, die gruen (oder zumindest farbig :) sind, muessen ausgeschrieben werden, die weissen Teile koennen weggelassen werden.

Wenn Sie farbenblind sind oder nur einen S/W-Monitor haben, koennen Sie das auch aus dem Hilfstext zu dem entsprechenden Befehl ersehen, dort ist zusaetzlich ersichtlich, wie weit die Optionsnamen abgekuerzt werden koennen. Die Teile, die ausgeschrieben werden muessen, sind dort in Gross-schrift gehalten.

## 1.7 CLI / Fehlermeldungen

### Fehlermeldungen

Nach der Eingabe analysiert AMMS die Befehlszeile auf ihre Richtigkeit und gibt ggf. eine Fehlermeldung aus. Die Stelle, an der AMMS den Fehler in der Zeile entdeckt hat, wird durch ein '^'-Zeichen markiert.

Beispiel: Eingabe eines falschen (unbekannten) Befehls SHOW GURKE

```
AMMS> SHOW GURKE
-> SHOW GURKE          <-- Die Eingabe nochmal
->      ^              <-- Fehlerstelle
-> ?Unbekannter Befehl. <-- Fehlergrund
```

Den Befehl SHOW GURKE gibt es natuerlich nicht, wohl aber andere Befehle, die mit SHOW anfangen. Deswegen wurde erst das Wort GURKE als falsch erkannt und markiert. In seltenen Faellen kann es vorkommen, dass AMMS auf den ersten Blick eine falsche Fehlermeldung ausgibt oder eine vermeintlich falsche Stelle als fehlerhaft markiert. Das haengt mit der internen Verarbeitung der Befehlszeile zusammen - nobody ist perfect!

Beispiel: Eingabe eines vorhandenen Befehls mit falscher Option

```
AMMS> SHOW USER/BLUB
-> SHOW USER/BLUB
```

```
->          ^
-> ?Unbekannte Option.
```

Sie sehen also, AMMS gibt sich Muehe den Fehler moeglichst klar auszudruecken :-)

Eine komplette Liste aller Fehlermeldungen und ihrer Bedeutung findet sich unter

```
Fehlerliste
.
```

## 1.8 CLI / Hilfstexte

Hilfstexte

-----

Zu jedem Befehl laesst sich ein Hilfstext abrufen, in dem seine Parameter und Optionen erklart werden. Um z.B. den Hilstext zum LIST USER-Befehl abzurufen, geben Sie einfach ein:

```
AMMS> HELP LIST USER
```

Eine Liste der Standardbefehle erhalten Sie einfach durch den ?-Befehl, mit der Eingabe:

```
AMMS> ?
```

## 1.9 CLI / Parametertypen

Parametertypen

-----

AMMS unterscheidet bei den Parametern eines Befehles fuenf verschiedene Arten:

**NAME** Ein Name eines Users, Files etc. Wenn er Leerzeichen enhaelt, muss er in Anfuhrungszeichen gesetzt werden.

Beispiele anhand des SHOW USER-Befehls, dessen 1. Parameter der Username ist:

Anzeige der Userdaten des Users SYSOP :

```
AMMS> SHOW USER SYSOP
```

Anzeige der Userdaten des Users THE KING :

```
AMMS> SHOW USER "THE KING"
```

**STRING** Eine Zeichenkette in Anfuhrungszeichen. Kommen in der Zeichenkette Anfuhrungszeichen vor, so muessen diese durch jeweils zwei Anfuhrungszeichen ersetzt werden. Statt einer Zeichenkette kann

auch ein Variablenname angegeben werden.

Beispiele anhand des ECHO-Befehls, dessen 1. Parameter ein String ist:

Ausgabe der Zeichenkette "HALLO":

```
AMMS> ECHO "HALLO"
-> HALLO
```

Ausgabe der Zeichenkette "HALLO "KING" SYSOP":

```
AMMS> ECHO "HALLO ""KING"" SYSOP"
-> HALLO "KING" SYSOP
```

Ausgabe des Inhalts der Variable A:

```
AMMS> ECHO A
-> (nichts, da die Variable nicht definiert ist)
```

Bei der Angabe einer Zeichenkette kann das letzte Anfuhrungszeichen weggelassen werden, sofern es sich um den letzten Parameter handelt.

**ZAHL** Ein direkter Zahlenwert, z.B. "123", ein Variablenname oder ein Integer-Ausdruck (siehe Integer-Ausdruecke). Die Angabe eines direkten Zahlenwertes geschieht normalerweise dezimal, durch voranstellen eines speziellen Zeichens vor die Zahl sind jedoch auch andere Zahlensysteme moeglich:

% Dualsystem, als Ziffern nur 0 und 1 erlaubt, z.B.:  
%1011 (Dezimal 11)

& Octalsystem, als Ziffern nur 0 bis 7 erlaubt, z.B.:  
&725 (Dezimal 469)

\$ Hexadezimalsystem, als Ziffern sind Zahlen von 0 bis 9 und Buchstaben von A bis F erlaubt, z.B.:  
\$A2E (Dezimal 2606).

Vor der Zahl darf ggf. noch ein "-" als Vorzeichen stehen.

Aufgrund der internen Verarbeitung sind nur Zahlen im Bereich von +2147483647 bis -2147483648 erlaubt.

**DATUM** Eine direkte Angabe eines Datums in der Form TAG-MONAT-JAHR, TAG.MONAT.JAHR oder <ZAHL>. Die einzelnen Teile sind:

TAG Zahl von 1..31 (je nach Monat)  
MONAT Monatsname/Kuerzel oder Zahl 1..12  
JAHR Vierstellig (z.B. 1994) oder zweistellig (z.B. 94)

Bei der Angabe von <ZAHL> wird der entsprechende Wert in Tagen zum aktuellen Datum addiert und ergibt so das entsprechende Datum.

Einige Beispiele fuer gueltige Datumsangaben:

```
4.Januar.1996
4.Jan.1996
4-Jan-96
4.1.1996
4.1.96
4-01-1996
```

```
100          <-- Aktuelles Datum + 100 Tage
```

**ZEIT** Eine direkte Angabe einer Zeit in der Form HH:MM, HH:MM:SS oder <ZAHL>. Die einzelnen Teile sind:

```
HH  Stunden
MM  Minuten
SS  Sekunden
```

Bei der Angabe von <ZAHL> wird der entsprechende Wert in Sekunden zur aktuellen Zeit addiert und ergibt so die entsprechende Zeit.

Einige Beispiele fuer gueltige Zeitangaben:

```
13:25:42
13:25
```

```
200          <-- Aktuelle Zeit + 200 Sekunden
```

## 1.10 CLI / Symbole (Variablen), Abkuerzungen

Symbole (Variablen) und Abkuerzungen

-----

AMMS bietet die Moeglichkeit, Variablen (auch Symbole genannt) zu definieren. Variablen sind quasi Platzhalter, die ueber ihren Namen angesprochen werden und einen bestimmten Inhalt haben. Variablen koennen durch eine Befehlszeile in der Form

```
AMMS> <NAME>=[<=><WERT>
```

definiert werden.

Das zweite Gleichheitszeichen wird dann benoetigt, wenn in einer Batchdatei eine Variable definiert werden soll, die auch ausserhalb der Batchdatei ihren Wert beibehalten soll. Solche Variablen werden "globale Variablen" genannt. Alle Variablen die in einer Batchdatei nur mit "=" definiert wurden, werden von AMMS nach Ende dieser geloescht. In der Befehlsebene sind ALLE Zuweisungen global. Wird <WERT> weggelassen, so wird die entsprechende Variable geloescht.

Variablen koennen sowohl Strings als auch Zahlen beinhalten, ihr "Typ" wird bei ihrer ersten Zuweisung bestimmt.

Alle Variablen die global sind, fungieren als ALIAS, d.h. nach der Eingabe einer Befehlszeile prueft AMMS, ob es sich bei dem ersten Wort um den

Namen einer globalen Variable handelt und ersetzt diesen ggf. durch ihren Inhalt. Dieses Ersetzen kann durch das Zeichen '\_' vor dem ersten Befehls- wort unterdrueckt werden, was fuer Batchdateien in jedem Fall sinnvoll ist, um ungewollte Effekte zu vermeiden.

Einige Beispiele:

Der Variable WHO den String "SHOW PORTS" zuweisen, nach der Eingabe wird AMMS dann auf die Eingabe von "WHO" genauso reagieren wie bei dem Aufruf des Befehls "SHOW PORTS":

```
AMMS> WHO="SHOW PORTS"
      ^   ^
      |   |_____ Variablenwert (String)
      |_____ Variablenname
```

```
AMMS> WHO
-> <Ausgabe der User im System>
```

Die Variable WHO wieder loeschen :

```
AMMS> WHO=
```

Der Variable TEST den Wert 99 zuweisen :

```
AMMS> TEST=99
      ^   ^
      |   |_____ Variablenwert (Zahl)
      |_____ Variablenname
```

Der Variable TEST das Ergebnis einer Berechnung zuweisen:

```
AMMS> TEST=33+5*7-10                (Womit TEST den Wert 58 haette)
```

Anzeigen aller definierten Variablen mit dem Befehl SHOW SYMBOL:

```
AMMS> SHOW SYMBOL
-> <LISTE DER VARIABLEN UND IHRES INHALTES>
```

## 1.11 CLI / Integer-Ausdruecke

### Integer-Ausdruecke

Statt der direkten Angabe einer Zahl bzw. eines Variablennamens koennen auch ganze Ausdruecke ueberall dort angegeben werden, wo eine Zahl zulaes- sig ist. Als Operatoren sind folgende Zeichen verwendbar, geordnet nach ihrer Bindungsstaerke:

OPERATOR	FUNKTION	BEISPIEL
**	Potenzieren	3 ** 2 = 9
*	Multiplikation	2 * 5 = 10
/	Division	8 / 4 = 2

%	Modulo	$9 \% 2 = 1$
+	Addition	$3 + 4 = 7$
-	Subtraktion	$9 - 5 = 4$
<<	Shift Left	$1 \ll 4 = 16$
>>	Shift Right	$128 \gg 2 = 32$
&	UND-Verknuepfung	$11 \& 7 = 3$
^	XOR-Verknuepfung	$15 \wedge 2 = 13$
	ODER-Verknuepfung	$8   4 = 12$

Die Bindungsstaerke gibt an, in welcher Reihenfolge die Operatoren bearbeitet werden. So wird z.B. bei  $3**2*4$  zuerst das  $3**2$  berechnet, das Ergebnis davon wird mit 4 multipliziert. Um diese Reihenfolge ausser Kraft zu setzen, koennen Klammern verwendet werden. Wuerde z.B.  $3**(2*4)$  angegeben, so wird zuerst  $2*4$  berechnet und danach die 3 mit diesem Wert potenziert.

**WICHTIG:** Innerhalb des Ausdrucks duerfen keine Spaces vorkommen!

Beispiele:

```
AMMS> A=12+3*4
AMMS> SHOW SYMBOL A
-> A == 24
```

```
AMMS> A=2*12-2*3
AMMS> SHOW SYMBOL A
-> A == 18
```

## 1.12 CLI / Variablensubstitution

### Variablensubstitution

-----

Bevor eine eingegebene Befehlszeile weiter ausgewertet wird, ersetzt AMMS alle Variablen, die zwischen den Zeichen "'" stehen durch ihren Inhalt. Dadurch lassen sich mit Hilfe von Variablen beliebige Befehlszeilen zusammenbauen und ausfuehren.

Beispiel:

```
AMMS> VAR1="SHOW"           <-- Zuweisung Variable VAR1
AMMS> VAR2="USER"          <-- Zuweisung Variable VAR2
AMMS> VAR3="SYSOP"         <-- Zuweisung Variable VAR3
AMMS> 'VAR1' 'VAR2' 'VAR3' <-- AMMS erzeugt : "SHOW USER SYSOP"
-> <USERLISTE MIT USER SYSOP>
```

Soll das Zeichen "'" z.B. in einer Ausgabe verwendet werden, OHNE das AMMS es beachtet, so muss es doppelt geschrieben werden. Beispiel:

```
AMMS> ECHO "Wie geht's Dir ?"
-> Wie geht's Dir ?
```

## 1.13 CLI / Zugriffsrechte Uebersicht

### Zugriffsrechte Uebersicht

-----

Zugriffsrechte werden unter AMMS auf zwei verschiedene Arten vergeben: Mit dem Userlevel (im Folgenden kurz "Level" genannt) und den 16 Zugriffsflags (auch "Acc" genannt).

Ein User hat dann Zugriff auf einen Befehl oder eine Option eines Befehles, wenn er entweder den entsprechenden Level, die entsprechenden Zugriffsflags oder beides hat.

Innerhalb von Batchdateien, die vom User als Befehl gestartet werden, koennen dem User fuer die Laufzeit der Batchdatei ein hoeherer Level und/oder zusaetzliche Zugriffsflags gesetzt werden.

Die Zugriffsrechte eines Users koennen mit dem Befehl EDIT USER veraendert werden, die Zugriffsdaten eines Befehles koennen mit dem Befehl EDIT COMMAND veraendert werden. Genauere Beschreibungen finden sich in den jeweiligen Hilfstexten zu den Befehlen.

## 1.14 CLI / Zugriffsrechte durch Level

### Zugriffsrechte durch Level

-----

Der Level ist eine Zahl zwischen 0 und 9999. Der Level 0 steht fuer den minimalen Zugriff und der Level 9999 fuer den maximalen Zugriff. Fuer jeden Befehl kann festgelegt werden, ab welchem Level der User darauf Zugriff hat.

Ein einfaches Beispiel ist ein Capture der Daten des Befehles SHOW PORTS:

```
Befehl      : SHow          USERLEVEL   : 1000      UserAcc    : -----
              POrts          SYSOPLEVEL  : 9000      SysopAcc   : -----
```

Der Eintrag USERLEVEL gibt hier an, dass alle User mit einem Level ab 1000 Zugriff auf diesen Befehl haben.

Zusaetzlich zum "normalen" Userzugriff, kann man auf Befehle auch noch Sysopzugriff haben. Wie hier zu sehen, steht im Eintrag SYSOPLEVEL, dass alle User ab Level 9000 Sysopzugriff auf diesen Befehl haben.

Was ist nun der Unterschied zwischen Userzugriff und Sysopzugriff? Beim Befehl SHOW PORTS macht sich dieser Unterschied nur dadurch bemerkbar, dass man mit Sysopzugriff noch die Level der eingeloggten User sieht.

Fuer einen User mit Level 1000, sprich Userzugriff, wuerde die Ausgabe von

SHOW PORTS so aussehen:

```
Nr.  User          Port          Baud  Befehl
-----
 1   Sysop         con0          Lokal  SHOW PORTS
```

1 User online.

Fuer einen User mit Level 9000 oder hoeher, sprich Sysopzugriff, wuerde die Ausgabe so aussehen:

```
Nr.  User          Port          Baud  Befehl          Level
-----
 1   Sysop         con0          Lokal  SHOW PORTS      9999
```

1 User online.

## 1.15 CLI / Zugriffsrechte durch Zugriffsflags (Acc)

### Zugriffsrechte durch Zugriffsflags (Acc)

Der Acc besteht aus 16 Zugriffsflags, 0-9 und a-f, mit denen ebenfalls User-und Sysopzugriff gesteuert werden koennen. Ein User hat dann den entsprechenden Zugriff, wenn sein Acc mindestens ein Zugriffsflag aus dem Acc des Befehles enthaelt. Mit diesen Zugriffsflags kann man so die User noch in Zugriffsgruppen unterteilen.

Ein Beispiel dafuer ist z.B. der Befehl APPLICATION (Userantrag), hier ein Ausschnitt aus seinen Befehlsdaten:

```
Befehl   : APplication      Userlevel  : 9000      USERACC   : 0-----
                  Sysoplevel  : 9000      SysopAcc  : -----
```

Man sieht also, dass auf diesen Befehl nur User mit einem Level ab 9000 Zugriff haetten. Wie kann jedoch der Gast mit Level 1000 trotzdem diesen Befehl aufrufen um einen Userantrag zu stellen?

Hierzu die Daten des Users GAST aus dem Befehl LIST USER:

```
Gast          1000  0----- ...
```

Wie man sieht, ist sowohl beim Gast als auch im USERACC-Eintrag vom APPLICATION-Befehl das Zugriffsflag 0 gesetzt, womit der User trotz seines zu kleinen Levels Zugriff auf den APPLICATION-Befehl hat.

## 1.16 Zugriffsvergabe fuer Optionen eines Befehles

### Zugriffsvergabe fuer Befehlsoptionen

Fuer die verschiedenen Optionen eines Befehls koennen ebenfalls



individuell die Zugriffe eingestellt werden.

Als Beispiel hierfuer die Befehlsdaten des SET EMULATION-Befehls, der zum Einstellen der Emulation dient:

```

Befehl      : SET              Userlevel   : 1000      UserAcc    : -----
              Emulation        Sysoplevel   : 9000      SysopAcc   : -----
Flags       : -----         Batchlevel  : 0        BatchAcc   : -----

1. Option   : Call           Level      : 1000      OptAcc    : -----
2. Option   : Old            Level      : 1000      OptAcc    : -----
3. Option   : User          Level      : 9000      OptAcc    : -----
4. Option   : Quiet         Level      : 0          OptAcc    : -----
5. Option   :                Level      : 0          OptAcc    : -----
6. Option   :                Level      : 0          OptAcc    : -----

```

Man sieht hier, das z.B. die Option USER (Zur Angabe eines Usernamens) nur fuer User mit einem Level 9000 oder hoeher verwendbar ist. Optionen fuer die kein individueller Zugriff festgelegt werden soll, werden mit Level 0 eingetragen.

## 1.17 Zugriffsrechte in Befehls-Batchdateien

### Zugriffsrechte in Befehls-Batchdateien

Bei einigen AMMS-Befehlen handelt es sich nicht um Binaerfiles, sondern um Batchdateien. In solchen Batchdateien werden oft Befehle benoetigt, auf die normalerweise nur der Sysop Zugriff hat. Dabei kann es sich z.B. um Befehle fuer Doors zu diversen Onlinegames o.ae. handeln. Um dieses Problem zu loesen, ist es moeglich, dem User `_FUER DIE LAUFZEIT DER BATCHDATEI_` einen hoeheren Level und zusaetzliche Zugriffsflags im Acc zu geben.

Ein Beispiel hierfuer ist der Befehl INFO SYSTEM. Dieser Befehl ist eine AMMS-Batchdatei und gibt u.a. auch den freien Speicherplatz des Rechners aus. Dazu wird ueber den AMMS-Befehl EXTERN der Amiga-DOS Befehl AVAIL aufgerufen. Der Befehl EXTERN darf NUR dem Sysop direkt zugaenglich sein, da man damit logischerweise eine Menge Unheil anrichten koennte. Damit nun aber der Befehl innerhalb der INFO SYSTEM-Batchdatei, auch vom normalen User, ausgefuehrt werden kann, bekommt der User waehrend der Ausfuehrung dieser Batchdatei das Zugriffsflag 6 gesetzt.

Ein Ausschnitt aus den Befehlsdaten des EXTERN-Befehles:

```

Befehl      : EXtern          Userlevel   : 9999      UserAcc    : -----6-----
              Sysoplevel      : 9999      SysopAcc   : -----

```

Man sieht, dass der Befehl nur Usern mit dem Level 9999 oder dem Zugriffsflag 6 zugaenglich ist.

Dazu die Daten des INFO SYSTEM-Befehles:

```

Befehl      : Info           Userlevel   : 0          UserAcc    : -----
              System         Sysoplevel   : 9999      SysopAcc   : -----

```

Flags : -----78            Batchlevel : 0            BatchAcc : -----6-----

Hier wurde bei BatchAcc das Flag 6 gesetzt, was bewirkt, dass der User fuer die Ausfuehrungszeit der Batch das Zugriffsflag 6 besitzt. Eine alternative Loesung waere hier gewesen, statt dem Zugriffsflag den Batchlevel auf 9999 zu setzen, so dass der User fuer die Ausfuehrungszeit der Batch den Level 9999 hat.

## 1.18 CLI / Liste der Fehlermeldungen

Fehlermeldungen, ihre Bedeutungen und Tips zur Beseitigung

-----

?Bufferfehler.

Irgendwo ist eine Zeile oder ein String zu lang geworden, so dass der Eingabebuffer nicht mehr ausreicht.

-> Evtl. kuerzere Symbolnamen und Labelnamen verwenden.

?Unguelteiger Symbolname.

In einem Symbolnamen sind nur die Zeichen '0'..'9', 'A'..'Z', 'a'..'z' und '\_' erlaubt.

-> Nur die angegebenen Zeichen im Symbolnamen verwenden.

?Unguelteiger Befehlsname.

In Befehlsnamen sind nur die Zeichen 'A'..'Z', 'a..'z' und '?' erlaubt.

-> Befehlsnamen korrigieren, Liste aller Befehle mit '?' abrufbar.

?Unguelteiger Optionsname.

In einem Optionsnamen sind nur die Zeichen 'A'..'Z', 'a..'z' und '?' erlaubt.

-> Optionsnamen korrigieren, zulaessige Optionen mit HELP <BEFEHL> abrufbar.

?Unbekanntes Symbol.

Das angegebene Symbol existiert nicht, bzw. ist nicht definiert.

-> Symbolnamen korrigieren, Liste aller definierten Symbole mit SHOW SYMBOL abrufbar.

?Ersetzen zu komplex.

Die Symbolersetzung beim Auswerten der Befehlszeile ist zu komplex.

-> Auf unzuulaessige Rekursionen bei der Ersetzung achten.

?Darauf hast Du keinen Zugriff :-).

Es wurde versucht, einen Befehl oder eine Option zu benutzen, auf die Sie keinen Zugriff haben.

-> Level bzw. ComAcc aendern, bei Befehls-Batchdateien evtl. Batchlevel bzw. BatAcc aendern.

?Unbekannter Befehl.

Der angegebene Befehl existiert nicht.

-> Befehlsnamen korrigieren, Liste aller Befehle mit "?" abrufbar.

?Unbekannte Option.

Die angegebene Option existiert bei dem angegebenen Befehl nicht.

-> Optionsnamen korrigieren, zulaessige Optionen mit HELP <BEFEHL> abrufbar.

?Ladefehler.

Der angegebene Befehl konnte nicht geladen werden.

-> Mit EDIT COMMAND pruefen, ob richtiger Filename fuer Befehl eingetragen.

?Unguelteige Zahl.

Die angegebene Zahl ist fehlerhaft.

-> Zahl korrigieren. Zulaessige Ziffern siehe unter  
Parametertypen  
.

?Unguelteiges Datum.

Das angegebene Datum ist fehlerhaft.

-> Datum korrigieren. Zulaessige Angaben siehe unter  
Parametertypen  
.

?Unguelteige Zeit.

Die angegebene Zeit ist fehlerhaft.

-> Zeit korrigieren. Zulaessige Angaben siehe unter  
Parametertypen  
.

?Name zu lang.

Der angegebene Name ist im Rahmen des Befehls zu lang.

-> Kuerzeren Namen angeben.

?String zu lang.

Der angegebene String ist im Rahmen des Befehls zu lang.

-> Kuerzeren String angeben.

?Zahl zu klein/gross.

Die angegebene Zahl ist im Rahmen des Befehls zu klein oder zu gross.

-> Zahl im gueltigen Bereich waehlen, evtl. mit HELP <BEFEHL> nach-  
sehen, was erlaubt ist.

?Zu viele Parameter.

Es wurden zuviele Parameter fuer den Befehl angegeben.

---

-> Weniger Parameter angeben, evtl. mit HELP <BEFEHL> nachsehen, was bei dem Befehl als Parameter angegeben werden kann. Usernamen mit einem Leerzeichen muessen zwischen "" angegeben werden!

?Ausdruck fehlt.

Bei einer Angabe in der Form "Option=..." fehlt hinter dem "=" der Ausdruck.

-> Ausdruck angeben, evtl. mit HELP <BEFEHL> Hilfe zu den Optionen abrufen.

?Kein Ausdruck erlaubt.

Bei der Option ist kein "=" erlaubt, sie ist nur ein "Schalter".

-> "=" loeschen, evtl. mit HELP <BEFEHL> Hilfe zu den Optionen abrufen.

?Fehler beim setzen des Symbols.

Das angegebene Symbol kann nicht gesetzt werden.

-> Vermutlich wurden in einer Batchdatei zu viele Symbole verwendet.

?Fehler beim loeschen des Symbols.

Das angegebene Symbol konnte nicht geloescht werden.

-> Sollte NIE auftreten, Fehler in der internen Symbolverwaltung !

?Zu wenig Stack.

Es ist zu wenig Stack vorhanden, um eine Batchdatei noch weiter zu schachteln.

-> In die AmigaDOS Startbatch fuer AMMS den Befehl STACK <STACKSIZE> einbauen. Ein sinnvoller Wert fuer <STACKSIZE> ist z.B. 8000, der der Standardwert ist 4000.

?Klammerfehler.

Bei einem numerischen Ausdruck stimmt die Anzahl der geoeffneten Klammern nicht mit der Anzahl der geschlossenen Klammern ueberein.

-> Ausdruck ueberpruefen, Klammern korrigieren.

?Unbekannter Operator.

In einem numerischen Ausdruck wurde ein unbekannter Operator verwendet.

-> Ausdruck ueberpruefen. Zulaessige Operatoren siehe unter

Integer-Ausdruecke

.

?Division durch Null.

In einem numerischen Ausdruck trat bei der Berechnung eine Division durch Null auf.

-> Fehler durch Symbole mit dem Inhalt 0 lassen sich durch Abfragen vor der entsprechenden Berechnung vermeiden.

---

<BEFEHL>: Nur in Batchdateien erlaubt !  
Der angegebene Befehl laesst sich nur in Batchdateien verwenden.

<BEFEHL>: Label <LABEL> existiert nicht !  
Das angegebene Label konnte in der Batchdatei nicht gefunden werden.

-> Labelnamen ueberpruefen.

<BEFEHL>: Batchstack voll !  
Es wurden zuviele GOSUBs geschachtelt.

-> Versuchen, weniger zu schachteln.

<BEFEHL>: Zu wenig Speicherplatz frei !  
Es ist zu wenig Speicher vorhanden, um den angegebenen Befehl ordnungsgemaess auszufuehren.

-> RAM-Erweiterung kaufen.

<BEFEHL>: Kann File <FILE> nicht laden !  
Der angegebene Befehl konnte das angegebene File nicht laden.

-> Pfad/File ueberpruefen.

<BEFEHL>: Kann File <FILE> nicht speichern !  
Der angegebene Befehl konnte das angegebene File nicht speichern.

-> Pfad/File ueberpruefen.

<BEFEHL>: File <FILE> zu lang !  
Der angegebene Befehl konnte das angegebene File nicht vollstaendig laden, da es zu lang ist.

-> Einige Befehle haben Beschraenkungen der maximlaen Filegroesse, evtl. mehrere Befehle hintereinander aufrufen.

<BEFEHL>: User "... " unbekannt !  
Der angegebene User konnte nicht gefunden werden.

-> Usernamen pruefen, evtl. mit LIST USER nachsehen, welche User vorhanden sind.

---