

# GINA Makrosprache

Version 1.2.0.0



rd electronic GmbH  
Zweigstelle Dresden  
Bernhardstraße 70  
01187 Dresden  
Tel. +49 351-6563-56-4  
Internet: <http://www.rd-electronic.de>

---

Dokumentnummer:	GINAMACDES
Ausgabe:	6.0.0.1
Status:	Release
letzte Änderung:	30. 03. 2021
erstellt am:	09. 04. 2010
Autor	SSc
Macro Version	1.2.0.0

## Rechtliche Hinweise

DIESES DOKUMENT ENTHÄLT INFORMATIONEN, WELCHE DURCH DAS URHEBERRECHT GESCHÜTZT SIND. KEIN TEIL DIESES DOKUMENTES DARF OHNE SCHRIFTLICHE GENEHMIGUNG DES AUTORS IN IRGEND EINER FORM, FOTOKOPIE, MICROFILM, ODER MIT EINEM ANDEREN VERFAHREN, AUCH NICHT FÜR ZWECKE DER UNTERRICHTSGESTALTUNG, REPRODUZIERT ODER UNTER VERWENDUNG ELEKTRONISCHER SYSTEME VERARBEITET, VERVIELFÄLTIGT ODER VERBREITET WERDEN.

DER AUTOR BEHÄLT SICH DAS RECHT VOR, DIE IN DIESEM DOKUMENT ENTHALTENEN INFORMATIONEN OHNE ANKÜNDIGUNG ZU ÄNDERN. DER ANWENDER IST VERPFLICHTET, SICH ÜBER DEN GÜLTIGEN STAND ZU INFORMIEREN.

OBWOHL BEI DER ERSTELLUNG DIESES DOKUMENTES MIT GRÖSSTER SORGFALT VORGEHEND WURDE, SIND FEHLER DENNOCH NICHT GANZ AUSZUSCHLIESSEN. AUS DIESEM GRUND ÜBERNIMMT DER AUTOR KEINE HAFTUNG FÜR FEHLER, DIE IN DIESEM DOKUMENT ENTHALTEN SIND ODER FÜR ZUFÄLLIGE ODER FOLGESCHÄDEN IM ZUSAMMENHANG MIT DER ANWENDUNG DIESES DOKUMENTES.

## Änderungsübersicht

Version	Datum	bearbeitet von	Änderung
1.0	31.03.2010,	SSc	• Initial
2.0	09.04.2010	SSc	• Update
3.0	21.04.2010	SSc	• Update
4.0	04.03.2014	JWo	• Marken und Sprünge eingefügt
5.0	05.03.2014	JWo	• Update nach Review
6.0	05.03.2014	JWo	• Update nach Review
6.0.0.1	30.03.2021	SBa	• Firmendaten

## Inhalt

<b>1 Allgemeines</b>	<b>7</b>
1.1 Identifikation	7
1.2 Ziel	7
1.3 Referenzierte Dokumente	7
1.4 Definitionen und Abkürzungen	7
<b>2 Makro</b>	<b>8</b>
2.1 Gesamtübersicht	8
2.1.1 Prozeduren	8
2.1.2 Parameter	9
2.1.3 Markierung des Input-Parameter-Blocks	10
2.2 Syntax	10
2.2.1 Allgemein	10
2.2.2 Prozedurschlüssel	10
2.2.3 Prozedur-Parameter	10
2.2.4 Input-Parameterblock	11
2.3 Makro-Prozeduren	12
2.3.1 Parameter	12
2.3.1.1 VDDNAME	12
2.3.1.2 VDDID	12
2.3.1.3 JOBID	12
2.3.1.4 COMMOBJECTNAME	12
2.3.1.5 OPERATIONNAME	12
2.3.1.6 DEVICEFUNCNAME	12
2.3.1.7 DRIVERINSTANCENAME	12
2.3.1.8 DEVICEFUNCID	12
2.3.1.9 MODULENAME	12
2.3.1.10 ALIGNMENT	13
2.3.1.11 DITLANGUAGE	13
2.3.1.12 DRIVERID	13
2.3.1.13 TIME	13
2.3.1.14 DCDFILE	13
2.3.1.15 DRIVERFILE	13
2.3.1.16 TRANSITIONID	13
2.3.1.17 VERSION	13
2.3.1.18 LBNAME	13
2.3.1.19 COUNT	13
2.3.1.20 CooErr	14
2.3.1.21 RC	14
2.3.1.22 Qual	14
2.3.1.23 Grade	14
2.3.1.24 Code	14
2.3.2 Makro Funktionen	15
2.3.2.1 GAT_LOADDRIVER	15
2.3.2.2 GAT_UNLOADDRIVER	16
2.3.2.3 GAT_DELAY	17
2.3.2.4 GAT_LABEL	18
2.3.2.5 GAT_JUMP	19

2.3.2.6	GAT_JUMP_IF	20
2.3.2.7	GDI_INITIATE	22
2.3.2.8	GDI_CREATEFUNC OBJ	24
2.3.2.9	GDI_CREATECOMMOBJ	25
2.3.2.10	GDI_CREATEOPERATION	26
2.3.2.11	GDI_DELETEOPERATION	27
2.3.2.12	GDI_DELETEFUNC OBJ	28
2.3.2.13	GDI_DELETECOMMOBJ	29
2.3.2.14	GDI_READ	30
2.3.2.15	GDI_WRITE	31
2.3.2.16	GDI_EXECUTE	32
2.3.2.17	GDI_CONCLUDE	34
2.3.2.18	GDI_ABORT	35
2.3.2.19	GDI_IDENT	36
2.3.2.20	GDI_STATUS	37
2.3.2.21	GDI_TRANSITION	38
2.3.2.22	START_MACRO	39
2.3.2.23	STOP_MACRO	40
2.4	Backus-Naur-Form (BNF) - Definitionen	41
2.5	Makrodefinition (gesamt)	45
2.6	Besonderheiten der Parametrierung	46
2.6.1	Datentyp Union	46
2.6.2	Datentyp String	46
2.6.3	Datentyp Sequenz	46
2.6.4	Datentyp Enumeration	46
2.6.5	Datentyp Funktionsreferenz (FORefValue)	46
2.6.6	VD-Handle (VdHandle)	47
<b>3</b>	<b>Beispiel</b>	<b>48</b>

## Abbildungen und Tabellen

Tabelle 2.1:	Übersicht Makroprozeduren	8
Tabelle 2.2:	Übersicht Parameter der Makroprozeduren	9
Tabelle 2.3:	Übersicht Schlüsselworte für Input-Parameter-Block	10

## 1 Allgemeines

### 1.1 Identifikation

Dieses Dokument beschreibt Syntax und Semantik des von GINA verwendeten Makroformats.

Die Beschreibung der Syntax der Makrosprache erfolgt in der Backus-Naur-Form (BNF)<sup>1</sup>.

### 1.2 Ziel

Über ein vom Anwender definiertes Makro können Parameter des ASAM-GDI Treibers geschrieben, gelesen und Prozeduren aufgerufen werden. Mit der Hilfe eines Makros ist es also möglich, einen vorher definierten Arbeitsalgorithmus durch eine beliebige Anwendung, selbständig und zu jedem Zeitpunkt reproduzierbar ablaufen zu lassen.

### 1.3 Referenzierte Dokumente

- [1] ASAM GDI Part B Specification of the Interface of an ASAM GDI Driver Version 4.2 / 4.3 /ASAM GDI Generic Device Interface Part 1 of 2 Programmers Guide Base Version 4.5.0 Base Standard

### 1.4 Definitionen und Abkürzungen

GINA	– Generic Device Interface Analysis Tool
DCD	– Device Capability Definition
IDL	– Interface Definition Language
DIT	– Device Info Text
CR/LF	– carriage return, line feed

---

<sup>1</sup> Die Backus-Naur-Form oder Backus-Normalform (BNF) ist eine kompakte formale Metasyntax, welche zur Beschreibung kontextfreier Grammatiken (= Typ-2-Grammatiken vgl. Chomsky-Hierarchie ) benutzt wird. Hierzu zählt die Syntax gängiger höherer Programmiersprachen. Sie findet ebenfalls Verwendung bei der Notation von Befehlssätzen und Kommunikationsprotokollen.

## 2 Makro

### 2.1 Gesamtübersicht

#### 2.1.1 Prozeduren

Funktion	Beschreibung
START_MACRO	Kennzeichnet den Beginn eines Makros
STOP_MACRO	Kennzeichnet das Ende eines Makros
GAT_LOADDRIVER	Laden eines ASAM GDI Treibers
GAT_UNLOADDRIVER	Entladen eines ASAM GDI Treibers
GAT_DELAY	Wartezyklus
GAT_LABEL	Label
GAT_JUMP	Sprung
GAT_JUMP_IF	Bedingter Sprung
GDI_INITIATE	Aufruf von GDI_Initiate (siehe[1])
GDI_CREATEFUNCOBJ	Aufruf von GDI_CreateFuncObj (siehe[1])
GDI_CREATECOMMOBJ	Aufruf von GDI_CreateComObj (siehe[1])
GDI_CREATEOPERATION	Anlegen eines Referenzobjektes einer Operation.
GDI_DELETEOPERATION	Löschen eines Referenzobjektes einer Operation.
GDI_DELETEFUNCOBJ	Aufruf von GDI_DeleteFuncObj (siehe[1])
GDI_DELETECOMMOBJ	Aufruf von GDI_DeleteComObject (siehe[1])
GDI_READ	Aufruf von GDI_Read (siehe[1])
GDI_WRITE	Aufruf von GDI_Write (siehe[1])
GDI_EXECUTE	Aufruf von GDI_Execute (siehe[1])
GDI_CONCLUDE	Aufruf von GDI_Conclude (siehe[1])
GDI_ABORT	Aufruf von GDI_Abort (siehe[1])
GDI_IDENT	Aufruf von GDI_Identify (siehe[1])
GDI_STATUS	Aufruf von GDI_Status (siehe[1])
GDI_TRANSITION	Wechsel in eine neue Transitionstatus (siehe [1])

Tabelle 2.1: Übersicht Makroprozeduren



## 2.1.2 Parameter

Parameter-Schlüsselwort	Beschreibung
VDNAME	Name des VD-Handles.
MODULENAME	Name des jeweiligen Moduls.
VDID	Hat keine weitere Bedeutung innerhalb des Gina-Macros.
JOBID	Job-Id nach Definition in [1]
DEVICEFUNCNAME	Name der Devicefunction
DEVICEFUNCID	In DCD definierte Id der Devicefunction
COMMOBJECTNAME	In der DCD definierter Name des ComObjects.
OPERATIONNAME	In der DCD definierter Name der Operation.
ALIGNMENT	Beschreibt das Struct-Member-Alignment des GDI Treibers
DITLANGUAGE	Sprache der DIT
DRIVERID	ID der Treiberinstanz
TIME	Zeitangabe
DCDFILE	Name und Pfad der DCD
DRIVERFILE	Name und Pfad des GDI-Treibers
TRANSITIONID	Beschreibung der GDI-Transitionsfunktionen
VERSION	Makroversion
LBNAME	Labelname
COUNT	Anzahl der Sprünge
CooErr	Coordinator Fehlercode
RC	Return Code der GDI-Fehlerstruktur des Gerätetreibers.
Qual	Qual der GDI-Fehlerstruktur des Gerätetreibers.
Grade	Grade der GDI-Fehlerstruktur des Gerätetreibers.
Code	Code der GDI-Fehlerstruktur des Gerätetreibers.

Tabelle 2.2: Übersicht Parameter der Makroprozeduren

### 2.1.3 Markierung des Input-Parameter-Blocks

Block Parameter-Schlüsselwort	Beschreibung
BEGIN_INPUTPARAMETER	Kennzeichnet den Beginn eines Blocks mit Inputparametern.
END_INPUTPARAMETER	Kennzeichnet das Ende eines Blocks mit Inputparametern.
NO_INPUTPARAMETER	Keine Inputparameter vorhanden.

Tabelle 2.3: Übersicht Schlüsselworte für Input-Parameter-Block

## 2.2 Syntax

### 2.2.1 Allgemein

Die in diesem Dokument beschriebene Makrosprache (<MacroDefinition>) setzt sich aus einer Menge von definierten Prozeduren und den zugehörigen Parametern zusammen, welche der GDI-Treibersteuerung bzw. dem Austausch von Daten zwischen GDI-Treiber und Anwendung dienen. Die Anweisungen eines Makros werden von oben mit der ersten Anweisung beginnend, sequentiell nach unten abgearbeitet.

Eine Anweisung kann folgende Eintragungen enthalten:

- ein definiertes Prozedurschlüsselwort, das die auszuführende Funktion beschreibt
- die zugehörigen Parameter einschließlich einem Input-Parameterblocks, welcher die Übergabeparameter der aufzurufenden ASAM-GDI-API-Funktion definiert

Ein Makro startet immer mit der **START\_MACRO**-Anweisung, welche den Beginn eines Makros kennzeichnet. Das Ende eines Makros wird mit der Anweisung **STOP\_MACRO** gekennzeichnet. Zwischen diesen beiden Anweisungen kann eine unbestimmte Menge von Prozeduren definiert werden.

### 2.2.2 Prozedurschlüssel

Das Prozedurschlüsselwort beschreibt die auszuführende Funktion. Es ist eindeutig definiert (siehe Tabelle 2.1: Übersicht Makroprozeduren) und wird durch eckige Klammern [] begrenzt. Es steht am Beginn einer Anweisung gefolgt von zugehörigen Parametern und Beschreibungen.

### 2.2.3 Prozedur-Parameter

Die Parameter einer Anweisung folgen dem Schlüsselwort der Prozedur. Sie werden durch das sie anführende Schlüsselwort näher gekennzeichnet. Alle Parameter-Schlüsselworte sind verbindlich definiert (siehe Tabelle 2.2: Übersicht Parameter der Makroprozeduren). Die zu einem Parameter gehörenden Schlüsselworte sind in jedem Fall in ein „kleiner als“ / „größer als“ Zeichen „<“, „>“ einzuschließen.

#### 2.2.4 *Input-Parameterblock*

In diesem Block werden die Inputparameter der ASAM-GDI-API-Funktionen GDI\_CreateFuncObj, GDI\_Execute, GDI\_Write und GDI\_Initiate definiert. Der Beginn und das Ende eines Blocks werden durch **BEGIN\_INPUTPARAMETER** bzw. **END\_INPUTPARAMETER** gekennzeichnet. Innerhalb des Blocks beginnt die Parameterdefinition mit dem Namen des Parameters gefolgt von Leerzeichen und dem anschließenden Wert. Charakterwerte sind hierbei mit Hochkommas zu begrenzen. Das Schlüsselwort **NO\_INPUTPARAMETER** wird verwendet, wenn keine Input-Parameter vorhanden sind. Die Schlüsselworte **BEGIN\_INPUTPARAMETER**, **END\_INPUTPARAMETER** und **NO\_INPUTPARAMETER** werden durch eckige Klammern [] begrenzt.

## 2.3 Makro-Prozeduren

### 2.3.1 *Parameter*

#### 2.3.1.1 *VDNAME*

Name der VD-Handle Instanz. Er wird vom Anwender vergeben und ist eindeutig.

#### 2.3.1.2 *VDID*

Der Parameter ist zur internen Nutzung vorgesehen. Er wird nur im GDI-Engine-Makro-Format für GICO22 benutzt. Der Parameter hat keine weitere Bedeutung innerhalb des Gina-Macros.

#### 2.3.1.3 *JOBID*

Job-Id der ASAM GDI-API-Funktionen zur Festlegung von synchroner oder asynchroner Kommunikation.

#### 2.3.1.4 *COMMOBJECTNAME*

In der DCD festgelegte Name des Comobjects.

#### 2.3.1.5 *OPERATIONNAME*

In der DCD festgelegte Name der Operation.

#### 2.3.1.6 *DEVICEFUNCNAME*

Name der generierten Devicefunction-Instanz, welche vom Anwender festgelegt wird. Der Instanznamen muss eindeutig sein.

#### 2.3.1.7 *DRIVERINSTANCENAME*

Instanzname des GDI-Treibers. Der Instanzname muss eindeutig sein.

#### 2.3.1.8 *DEVICEFUNCID*

In der DCD festgelegte Id der Devicefunction.

#### 2.3.1.9 *MODULENAME*

Name des jeweiligen Moduls, welcher in der DCD festgelegt ist.

### *2.3.1.10 ALIGNMENT*

Beschreibt das Struct-Member-Alignment des GDI Treibers.

### *2.3.1.11 DITLANGUAGE*

Definiert die Sprache der DIT. Die Angabe hinsichtlich der Sprachdefinitionen erfolgt über die in [1] getroffenen Festlegungen.

### *2.3.1.12 DRIVERID*

ID der Treiberinstanz. Im GINA- Makro wird als Driver-ID der Instanznamen des GDI-Treibers bezeichnet, welcher sich aus einer Zeichenkette zusammensetzt.

### *2.3.1.13 TIME*

Zeitangabe in Millisekunden.

Der Parameter „INFINITE“ stoppt das Makro.

### *2.3.1.14 DCDFILE*

Name und Pfad der zu verwendenden DCD-Datei.

### *2.3.1.15 DRIVERFILE*

Name und Pfad des zu verwendenden GDI-Treibers.

### *2.3.1.16 TRANSITIONID*

Beschreibt die ASAM-GDI Transitionsfunktionen (siehe [1]) für den Übergang in einen neuen Transitionszustand.

### *2.3.1.17 VERSION*

Version des Makros (hier 1.2.0.0)

### *2.3.1.18 LBNAME*

Name des Labels (Sprungziel).

### *2.3.1.19 COUNT*

Anzahl der Sprünge.

#### *2.3.1.20 CooErr*

Coordinator Fehler Code.

#### *2.3.1.21 RC*

Return Code der GDI-Fehlerstruktur des Gerätetreibers.

#### *2.3.1.22 Qual*

Qual der GDI-Fehlerstruktur des Gerätetreibers.

#### *2.3.1.23 Grade*

Grade der GDI-Fehlerstruktur des Gerätetreibers.

#### *2.3.1.24 Code*

Code der GDI-Fehlerstruktur des Gerätetreibers.

## 2.3.2 Makro Funktionen

### 2.3.2.1 GAT\_LOADDRIVER

#### Prototype:

```

"[GAT_LOADDRIVER]" <Newline>
"<DRIVERFILE>" <Blank> <DriverName> <Newline>
"<DRIVERID>" <Blank> <DriverInstanceName> <Newline>
"<DCDFILE>" <Blank> <DCDName> <Newline>
"<DITLANGUAGE>" <Blank> <Language> <Newline>
"<ALIGNMENT>" <Blank> <NonZeroDecimal> <Newline>

```

#### Inputparameter:

<DRIVERFILE>	<DriverName>	Name und Pfad des ASAM-GDI Treibers.
<DCDFILE>	<DCDName>	Name und Pfad der DCD.
<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers (siehe 2.3.1.7 / 2.3.1.12)
<DITLANGUAGE>	<Language>	Zu verwendende Sprache in der entsprechenden DIT-Datei. Die Festlegungen hierzu werden in der DCD-Datei getroffen.
<ALIGNMENT>	<NonZeroDecimal>	Struktur Ausrichtung

#### Aufgabe:

Laden eines ASAM GDI Treibers sowie der zugehörigen DCD und DIT. Der GDI-Treiber und die DCD-Datei werden über Name und Pfad bestimmt. Die Festlegung DIT-Datei erfolgt über die in der DCD-Datei eingetragene Sprachdefinition.

### 2.3.2.2 GAT\_UNLOADDRIVER

**Prototype:**

```
"[GAT_UNLOADDRIVER]" <Newline>  
"<DRIVERID>"<Blank><DriverInstanceName><Newline>
```

**Inputparameter:**

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
------------	----------------------	-------------------------------

**Aufgabe:**

Entladen eines ASAM GDI Treibers sowie der zugehörigen DCD und DIT. Alle bisher angelegten Instanzen des GDI Treibers werden gelöscht.



### 2.3.2.3 GAT\_DELAY

**Prototype:**

```
"[ GAT_DELAY]" <Newline>  
"<TIME>"<Blank>[<Millisec>|"INFINITE"] <Newline>
```

**Inputparameter:**

<TIME>	<Millisec>	Wartezeit in Millisekunden
--------	------------	----------------------------

**Aufgabe:**

Durchführen eines Wartezyklus, dessen Dauer in Millisekunden definiert wird.

Der Parameter „INFINITE“ beschreibt einen unendlichen Wartezyklus. In diesem Fall muss die Weiterverarbeitung durch GINA erneut gestartet werden.

#### 2.3.2.4 GAT\_LABEL

**Prototype:**

```
“[GAT_LABEL]“ <Newline>  
“<LBNAME>“<Blank><Identifier><Newline>
```

**Inputparameter:**

<LBNAME>	<Identifier>	Labelname
----------	--------------	-----------

**Aufgabe:**

Markierung innerhalb des Ablaufs, die zum Beispiel als Sprungziel verwendet werden kann. Es wird keine Aktion ausgeführt. In der Spalte CooErr wird mit dem Rückgabewert 0 das Durchlaufen des Schrittes angezeigt. Dies hat keinen Einfluss auf die Auswertung von Sprungbedingungen.

### 2.3.2.5 GAT\_JUMP

**Prototype:**

```
“[GAT_JUMP]“ <Newline>  
“<LBNAME>“<Blank><Identifier><Newline>  
[“<COUNT>“<Blank>[<NonZeroDecimal><Decimaldigit>*<Newline>]’
```

**Inputparameter:**

<LBNAME>	<Identifier>	Sprungziel (Labelname)
Optional: <COUNT>	[<NonZeroDecimal><Decimaldigit>*]	Anzahl der Sprungwiederholungen

**Aufgabe:**

Springt zum angegebenen Sprungziel innerhalb des Makroablaufs und setzt den Ablauf am Sprungziel fort. Die Anzahl der Sprungwiederholungen bestimmt die Häufigkeit der Sprungausführungen. Erreicht der Zähler die Anzahl der Sprungwiederholungen, so wird nicht gesprungen sondern der Zähler auf 0 und der Rückgabewert -1 gesetzt. In diesem Fall wird mit dem nachfolgenden Makroschritt fortgesetzt. In der Spalte CooErr wird mit dem Rückgabewert 0 das Ausführen des Sprunges angezeigt, entsprechend zeigt der Wert -1 die Nichtausführung an. Der Rückgabewert hat keinen Einfluss auf die Auswertung von Sprungbedingungen.

## 2.3.2.6 GAT\_JUMP\_IF

### Prototype:

```

“[GAT_JUMP_IF]“ <Newline>
“<LBNAME>“<Blank><Identifier><Newline>
[“<COUNT>“<Blank>[<NonZeroDecimal><Decimaldigit>*]<Newline>]?
[“<CooErr>“<Blank><Operator>[“0“|“-“[“-“?<NonZeroDecimal><Decimaldigit>*]]<Newline>]?
[“<RC>“<Blank><Operator>[“0“|“-“[“-“?<NonZeroDecimal><Decimaldigit>*]]<Newline>]?
[“<Qual>“<Blank><Operator>[“0“|“-“[“-“?<NonZeroDecimal><Decimaldigit>*]]<Newline>]?
[“<Grade>“<Blank><Operator>[“0“|“-“[“-“?<NonZeroDecimal><Decimaldigit>*]]<Newline>]?
[“<Code>“<Blank><Operator>[“0“|“-“[“-“?<NonZeroDecimal><Decimaldigit>*]]<Newline>]?

```

### Inputparameter:

<LBNAME>	<Identifier>	Sprungziel (Labelname)
Optional: <COUNT>	[<NonZeroDecimal><Decimaldigit>*]	Anzahl der Sprungwiederholungen
Optional: <CooErr>	<Operator> [“0“ “-“[“-“?<NonZeroDecimal><Decimaldigit>*]]	Vergleichsfunktion Vergleichswert für den Rückgabewert „CooErr“ des zuletzt ausgeführten Kommandos
Optional: <RC>	<Operator> [“0“ “-“[“-“?<NonZeroDecimal><Decimaldigit>*]]	Vergleichsfunktion Vergleichswert für den Rückgabewert „RC“ des zuletzt ausgeführten Kommandos
Optional: <Qual>	<Operator> [“0“ “-“[“-“?<NonZeroDecimal><Decimaldigit>*]]	Vergleichsfunktion Vergleichswert für den Rückgabewert „Qual“ des zuletzt ausgeführten Kommandos
Optional: <Grade>	<Operator> [“0“ “-“[“-“?<NonZeroDecimal><Decimaldigit>*]]	Vergleichsfunktion Vergleichswert für den Rückgabewert „Grade“ des zuletzt ausgeführten Kommandos
Optional: <Code>	<Operator> [“0“ “-“[“-“?<NonZeroDecimal><Decimaldigit>*]]	Vergleichsfunktion Vergleichswert für den Rückgabewert „Code“ des zuletzt ausgeführten Kommandos

### Aufgabe:

Springt zum angegebenen Sprungziel innerhalb des Makroablaufs, wenn die parametrisierte Bedingung erfüllt ist und setzt den Ablauf am Sprungziel fort. Der Befehl prüft den zuletzt

gesetzten Rückgabewert (ohne GAT\_LABEL, GAT\_JUMP und GAT\_JUMP\_IF) bestehend aus "CooErr", "RC", "Qual", "Grade" und "Code". Die Anzahl der Sprungwiederholungen bestimmt die Häufigkeit der Sprungausführungen. Erreicht die aktuelle Anzahl von Durchläufen die Anzahl der Sprungwiederholungen, so wird nicht gesprungen sondern die aktuelle Anzahl auf 0 und der Rückgabewert -1 gesetzt. In diesem Fall wird mit dem nachfolgenden Makroschritt fortgesetzt. In der Spalte CooErr wird mit dem Rückgabewert 0 das Ausführen des Sprunges angezeigt, entsprechend zeigt der Wert -1 die Nichtausführung an. Der Rückgabewert hat keinen Einfluss auf die Auswertung von Sprungbedingungen.

## 2.3.2.7 GDI\_INITIATE

### Prototype:

```

“[GDI_INITIATE]” <Newline>
“<DRIVERID>”<Blank><DriverInstanceName><Newline>
“<VDNAME>”<Blank><VDName><Newline>
“<MODULENAME>”<Blank><ModulName><Newline>
[“<VDID>”<Blank><VDId><Newline>]?
“<JOBID>”<Blank><JobId><Newline>
[[“[BEGIN_INPUTPARAMETER]” <Newline>
<SetParameter><Newline>
“[END_INPUTPARAMETER]”]
|”[NO_INPUTPARAMETER]”] <Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name der zu erzeugenden VD-Handle-Instanz.
<VDID>	<VDId>	Keine Bedeutung im GINA-Makro.
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers
<MODULENAME>	<ModulName>	Name des in der DCD definierten Moduls

Parameterblock	<SetParameter>	Input Parameter der ASAM GDI API
----------------	----------------	----------------------------------

## Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_Initiate.

Durch GDI\_Initiate wird ein VD-Handle des Moduls <ModulName> angelegt, welches eine interne <VDId> erhält.

## 2.3.2.8 GDI\_CREATEFUNC OBJ

### Prototype:

```

“[GDI_CREATEFUNC OBJ]” <Newline>
“<DRIVERID>”<Blank><DriverInstanceName><Newline>
“<VDNAME>”<Blank><VDName><Newline>
“<JOBID>”<Blank><JobId><Newline>
“<DEVICEFUNCNAME>”<Blank><DevicefuncName><Newline>
“<DEVICEFUNCID>”<Blank><InterfaceId><Newline>
[[“[BEGIN_INPUTPARAMETER]”<Blank><Newline>
<SetParameter><Newline>
“[END_INPUTPARAMETER]”]
|”[NO_INPUTPARAMETER]”] <Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Instanzname der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<DEVICEFUNCID>	<InterfaceId>	In der DCD definierte ID der Gerätefunktion.
Parameterblock	<SetParameter>	Input Parameter der ASAM GDI API

### Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_CreateFuncObj.

Durch GDI\_CreateFuncObj wird eine Instanz mit der in der DCD definierten <InterfaceId> unter dem mit <VDName> angegebenen VD-Handle angelegt.



## 2.3.2.9 GDI\_CREATECOMMOBJ

### Prototype:

```

"[GDI_CREATECOMMOBJ]" <Newline>
"<DRIVERID>"<Blank><DriverInstanceName><Newline>
"<VDNAME>"<Blank><VDName><Newline>
"<JOBID>"<Blank><JobId><Newline>
"<DEVICEFUNCNAME>"<Blank><DevicefuncName><Newline>
"<COMMOBJECTNAME>"<Blank><COName><Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Instanzname der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<COMMOBJECTNAME>	<COName>	In der DCD definierter Name des Com-Objektes

### Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_CreateComObj.

Durch GDI\_CreateComObj wird ein ComObject mit dem in der DCD definierten <COName> unter dem mit <VDName> angegebenen VD-Handle und dem unter <DevicefuncName> angegebenen Func-Handle angelegt.

## 2.3.2.10 GDI\_CREATEOPERATION

### Prototype:

```

"[GDI_CREATEOPERATION]" <Newline>
"<DRIVERID>"<Blank><DriverInstanceName><Newline>
"<VDNAME>"<Blank><VDName><Newline>
"<JOBID>"<Blank><JobId><Newline>
"<DEVICEFUNCNAME>"<Blank><DevicefuncName><Newline>
"<OPERATIONNAME>"<Blank><OPName><Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Name der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<OPERATIONNAME>	<OPName>	In der DCD definierter Name der Operation

### Aufgabe:

Anlegen eines Referenzobjektes einer Operation im Koordinator.

.

## 2.3.2.11 GDI\_DELETEOPERATION

### Prototype:

```

“[GDI_DELETEOPERATION]” <Newline>
“<DRIVERID>”<Blank><DriverInstanceName><Newline>
“<VDNAME>”<Blank><VDName><Newline>
“<JOBID>”<Blank><JobId><Newline>
“<DEVICEFUNCNAME>”<Blank><DevicefuncName><Newline>
“<OPERATIONNAME>”<Blank><OPName><Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Name der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<OPERATIONNAME>	<OPName>	In der DCD definierter Name der Operation

### Aufgabe:

Löschen des jeweiligen Referenzobjektes einer Operation.

## 2.3.2.12 GDI\_DELETEFUNCOBJ

### Prototype:

```

"[GDI_DELETEFUNCOBJ]" <Newline>
"<DRIVERID>"<Blank><DriverInstanceName><Newline>
"<VDNAME>"<Blank><VDName><Newline>
"<JOBID>"<Blank><JobId><Newline>
"<DEVICEFUNCNAME>"<Blank><DevicefuncName><Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Instanzname der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation

### Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_DeleteFuncObj.

## 2.3.2.13 GDI\_DELETECOMMOBJ

### Prototype:

```

"[GDI_DELETECOMMOBJ]" <Newline>
"<DRIVERID>"<Blank><DriverInstanceName><Newline>
"<VDNAME>"<Blank><VDName><Newline>
"<JOBID>"<Blank><JobId><Newline>
"<DEVICEFUNCNAME>"<Blank><DevicefuncName><Newline>
"<COMMOBJECTNAME>"<Blank><COName><Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Instanzname der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<COMMOBJECTNAME>	<COName>	In der DCD definierter Name des Com-Objektes

### Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_DeleteComObj.

Durch GDI\_DeleteComObj wird das unter <COName> angegebene ComObject der entsprechenden Devicefunction (<DevicefuncName>) gelöscht.

## 2.3.2.14 GDI\_READ

### Prototype:

```

“[GDI_READ]” <Newline>
“<DRIVERID>”<Blank><DriverInstanceName><Newline>
“<VDNAME>”<Blank><VDName><Newline>
“<JOBID>”<Blank><JobId><Newline>
“<DEVICEFUNCNAME>”<Blank><DevicefuncName><Newline>
“<COMMOBJECTNAME>”<Blank><COName><Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Instanzname der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<COMMOBJECTNAME>	<COName>	In der DCD definierter Name des Com-Objektes

### Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_Read.

Lesen des Outputparameters des unter <COName> deklarierten ComObjects, welches Mitglied der unter <DevicefuncName> angegebenen Devicefunction ist.

## 2.3.2.15 GDI\_WRITE

### Prototype:

```

“[GDI_WRITE]” <Newline>
“<DRIVERID>”<Blank><DriverInstanceName><Newline>
“<VDNAME>”<Blank><VDName><Newline>
“<JOBID>”<Blank><JobId><Newline>
“<DEVICEFUNCNAME>”<Blank><DevicefuncName><Newline>
“<COMMOBJECTNAME>”<Blank><COName><Newline>
“[BEGIN_INPUTPARAMETER]” <Newline>
<SetParameter><Newline>
“[END_INPUTPARAMETER]” <Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Instanzname der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<COMMOBJECTNAME>	<COName>	In der DCD definierter Name des Com-Objektes
Parameterblock	<SetParameter>	Input Parameter der ASAM GDI API

### Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_Write.

Schreiben der im Parameterblock gesetzten Inputparameter des unter <COName> deklarierten ComObjects, welches Mitglied der unter <DevicefuncName> angegebenen Devicefunction ist.

## 2.3.2.16 GDI\_EXECUTE

### Prototype:

```

“[GDI_EXECUTE]” <Newline>
“<DRIVERID>”<Blank><DriverInstanceName><Newline>
“<VDNAME>”<Blank><VDName><Newline>
“<JOBID>”<Blank><JobId><Newline>
“<DEVICEFUNCNAME>”<Blank><DevicefuncName><Newline>
“<OPERATIONNAME>”<Blank><OPName><Newline>
[[“[BEGIN_INPUTPARAMETER]” <Newline>
<SetParameter>|<TransitionVD><Newline>
“[END_INPUTPARAMETER]”]
|”[NO_INPUTPARAMETER]”] <Newline>

```

### Inputparameter:

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<DEVICEFUNCNAME>	<DevicefuncName>	Vom Anwender definierter Name der Devicefunction
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<OPERATIONNAME>	<OPName>	In der DCD definierter Name der Operation
Parameterblock	<SetParameter> <TransitionVD>	Input Parameter der ASAM GDI API <TransitionVD> nur bei Transitionsfunktion

### Aufgabe:

Ausführen der ASAM GDI Funktion GDI\_Execute.



Schreiben der im Parameterblock gesetzten Inputparameter sowie Lesen der Outputparameter der unter <OPName> deklarierten Operation, welche Mitglied der unter <DevicefuncName> angegebenen Devicefunction ist.

### 2.3.2.17 GDI\_CONCLUDE

**Prototype:**

```
"[GDI_CONCLUDE]" <Newline>  
"<DRIVERID>"<Blank><DriverInstanceName><Newline>  
"<VDNAME>"<Blank><VDName><Newline>  
"<JOBID>"<Blank><JobId><Newline>
```

**Inputparameter:**

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation

**Aufgabe:**

Ausführen der ASAM GDI Funktion GDI\_Conclude.

Geordnetes löschen des unter <VDName> spezifizierten VD-Handles.

### 2.3.2.18 GDI\_ABORT

**Prototype:**

```
"[GDI_ABORT]"<Newline>  
"<DRIVERID>"<Blank><DriverInstanceName><Newline>  
"<VDNAME>"<Blank><VDName><Newline>
```

**Inputparameter:**

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles

**Aufgabe:**

Ausführen der ASAM GDI Funktion GDI\_Abort.

Löschen des unter <VDName> spezifizierten VD-Handles.

### 2.3.2.19 GDI\_IDENT

**Prototype:**

```
“[GDI_IDENT]“ <Newline>  
“<DRIVERID>”<Blank><DriverInstanceName><Newline>  
“<VDNAME>”<Blank><VDName><Newline>  
“<JOBID>”<Blank><JobId><Newline>
```

**Inputparameter:**

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation

**Aufgabe:**

Ausführen der ASAM GDI Funktion GDI\_Identify.

### 2.3.2.20 GDI\_STATUS

**Prototype:**

```
"[ GDI_STATUS]" <Newline>  
"<DRIVERID>"<Blank><DriverInstanceName><Newline>  
"<VDNAME>"<Blank><VDName><Newline>  
"<JOBID>"<Blank><JobId><Newline>
```

**Inputparameter:**

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation

**Aufgabe:**

Ausführen der ASAM GDI Funktion GDI\_Status.

### 2.3.2.21 GDI\_TRANSITION

**Prototype:**

```
"[ GDI_TRANSITION]"<Newline>  
"<DRIVERID>"<Blank><DriverInstanceName><Newline>  
"<VDNAME>"<Blank><VDName><Newline>  
"<JOBID>"<Blank><JobId><Newline>  
"<TRANSITIONID>"<Blank><Transition> <Newline>
```

**Inputparameter:**

<DRIVERID>	<DriverInstanceName>	Instanzname des GDI-Treibers.
<VDNAME>	<VDName>	VD-Name zugehörigen VD-Handles
<JOBID>	<JobId>	Job-Id zur Definition von synchroner oder asynchroner Kommunikation
<TRANSITIONID>	<Transition>	Transitionsfunktionen

**Aufgabe:**

Ausführen der ASAM GDI Transitionsfunktionen für das entsprechende unter <VDName> definierte VD.

### 2.3.2.22 *START\_MACRO*

**Prototype:**

```
“[START_MACRO]“ <Newline>  
“<Version>“<Blank><VERSION><Newline>
```

**Inputparameter:**

<Version>	<VERSION>	Versionsnummer Makros.	des
-----------	-----------	---------------------------	-----

**Aufgabe:**

Definiert Startpunkt des Makros.

### 2.3.2.23 *STOP\_MACRO*

**Prototype:**

“[STOP\_MACRO]“ <Newline>

**Inputparameter:**

--	--	--

**Aufgabe:**

Definiert Ende des Makros



## 2.4 Backus-Naur-Form (BNF) - Definitionen

<DriverName>	▶ [ <b>&lt;Letter&gt;&lt;Colon&gt;&lt;Backslash&gt;</b> <b>&lt;Letter&gt;&lt;Colon&gt;</b>   <b>&lt;Backslash&gt;</b>   <b>&lt;Letter&gt;</b>   <b>&lt;Slash&gt;</b>   <b>&lt;Letter&gt;</b>   <b>&lt;Underscore&gt;</b>   <b>&lt;Dot&gt;</b>   <b>&lt;Minus&gt;</b>   <b>&lt;Decimaldigit&gt;</b> ][ <b>&lt;Letter&gt;</b>   <b>&lt;Colon&gt;</b>   <b>&lt;Slash&gt;</b>   <b>&lt;Underscore&gt;</b>   <b>&lt;Backslash&gt;</b>   <b>&lt;Minus&gt;</b>   <b>&lt;Decimaldigit&gt;</b> ]* <b>&lt;Dot&gt;</b> [ <b>&lt;Letter&gt;</b> ]*
<Underscore>	▶ “ _ ”
<Backslash>	▶ “ \ ”
<Slash>	▶ “ / ”
<Colon>	▶ “ . ”
<Dot>	▶ “ . ”
<Minus>	▶ “ _ ”
<Operator>	▶ “ <   > “ “ = “ “ < > “ “ < = “ “ > = “
<DCDName>	▶ [ <b>&lt;Letter&gt;&lt;Colon&gt;&lt;Backslash&gt;</b> <b>&lt;Letter&gt;&lt;Colon&gt;</b>   <b>&lt;Backslash&gt;</b>   <b>&lt;Letter&gt;</b>   <b>&lt;Slash&gt;</b>   <b>&lt;Letter&gt;</b>   <b>&lt;Underscore&gt;</b>   <b>&lt;Dot&gt;</b>   <b>&lt;Minus&gt;</b>   <b>&lt;Decimaldigit&gt;</b> ][ <b>&lt;Letter&gt;</b>   <b>&lt;Colon&gt;</b>   <b>&lt;Slash&gt;</b>   <b>&lt;Underscore&gt;</b>   <b>&lt;Backslash&gt;</b>   <b>&lt;Minus&gt;</b>   <b>&lt;Decimaldigit&gt;</b> ]* <b>&lt;Dot&gt;</b> <b>&lt;Letter&gt;</b> ]*
<Language>	▶ [ <b>&lt;LowerCase&gt;&lt;LowerCase&gt;&lt;Underscore&gt;</b> <b>&lt;UpperCase&gt;&lt;UpperCase&gt;</b> ] [ <b>&lt;Letter&gt;&lt;Letter&gt;</b> ]
<VERSION>	▶ <b>&lt;NonZeroDecimal&gt;&lt;Decimaldigit&gt;* ” . ”</b> <b>&lt;Decimaldigit&gt;&lt;Decimaldigit&gt;* ” . ”</b> <b>&lt;Decimaldigit&gt;&lt;Decimaldigit&gt;* ” . ”</b> <b>&lt;Decimaldigit&gt;&lt;Decimaldigit&gt;* ” . ”</b>
<DriverInstanceName>	▶ <b>&lt;DriverName&gt;</b>   <b>&lt;Identifier&gt;</b>
<VDName>	▶ <b>&lt;Identifier&gt;</b>
<VDId>	▶ <b>&lt;Decimaldigit&gt;&lt;Decimaldigit&gt;*</b>
<JobId>	▶ <b>&lt;Decimaldigit&gt;&lt;Decimaldigit&gt;*</b>

<SetParameter>	<ul style="list-style-type: none"> <li>▶ &lt;Identifier&gt;&lt;Blank&gt; &lt;Value&gt;&lt;Newline&gt; [&lt;Identifier&gt;&lt;Blank&gt; &lt;Value&gt;&lt;Newline&gt;]*</li> </ul>
<Value>	<ul style="list-style-type: none"> <li>▶ [“-“? [&lt;NonZeroDecimal&gt;&lt;Decimaldigit&gt;*] [“-“? &lt;FloatValue&gt;  &lt;CharValue&gt;  &lt;StringValue&gt;  &lt;SeqValue&gt;  &lt;UnionValue&gt;  &lt;FORefValue&gt;  &lt;EnumValue&gt;  &lt;BoolValue&gt;  &lt;HexValue&gt;</li> </ul>
<CharValue>	<ul style="list-style-type: none"> <li>▶ “”&lt;Backslash&gt;?&lt;Character&gt;””</li> </ul>
<Character>	<ul style="list-style-type: none"> <li>▶ “ “   “!”   “#” to “&amp;”   “(“ to “~”</li> </ul>
<FloatValue>	<ul style="list-style-type: none"> <li>▶ [ [“.”&lt;Decimaldigit&gt;*] [&lt;Decimaldigit&gt;*[“.”&lt;Decimaldigit&gt;*]? ] ] [“e“ “E“][“+” “-“]?&lt;Decimaldigit&gt;*]?]</li> </ul>
<StringValue>	<ul style="list-style-type: none"> <li>▶ “”&lt;Character&gt;*””</li> </ul>
<SeqValue>	<ul style="list-style-type: none"> <li>▶ “Max”&lt;Blank&gt; [&lt;Decimaldigit&gt;][&lt;Decimaldigit&gt;* &lt;Newline&gt;”Length” &lt;Blank&gt; [&lt;NonZeroDecimal&gt;][&lt;Decimaldigit&gt;* &lt;Newline&gt;&lt;SetParameter&gt;</li> </ul>
<UnionValue>	<ul style="list-style-type: none"> <li>▶ “_identifierT1” &lt;Blank&gt; [“-“?&lt;Decimaldigit&gt;][&lt;Decimaldigit&gt;*] &lt;HexValue&gt; &lt;EnumValue&gt; &lt;CharValue&gt; &lt;BoolValue&gt;]&lt;Newline&gt; &lt;SetParameter&gt;</li> </ul>

<FORefValue>	► [[ <code>"&lt;DevicefuncName&gt;"</code> ]  <code>"NULLREF"</code> ] [ <code>","</code> ]  <code>"&lt;COInstanceName&gt;"</code> ]  [ <code>"NULLREF"</code> ]]] [ <code>&lt;InterfaceId&gt;":"</code> ] <code>&lt;Decimaldigit&gt;*</code>   <code>&lt;HexValue&gt;</code> ] [ <code>","</code> ] <code>&lt;COId&gt;":"</code> <code>&lt;Decimaldigit&gt;*</code> ?]
<BoolValue>	► <code>"TRUE"</code>   <code>"FALSE"</code>
<EnumValue>	► <code>&lt;Identifier&gt;</code>   <code>&lt;HexValue&gt;</code>   [[ <code>&lt;Decimaldigit&gt;</code> ][ <code>&lt;Decimaldigit&gt;*</code> ]]
<HexValue>	► <code>"0"</code> [ <code>"x"</code>   <code>"X"</code> ] <code>&lt;Hexdigit&gt;+</code>
<DevicefuncName>	► <code>&lt;Identifier&gt;</code>
<COInstanceName>	► <code>&lt;Identifier&gt;</code>
<OPInstanceName>	► <code>&lt;Identifier&gt;</code>
<Transition>	► [ <code>"START_DEFINITION"</code>   <code>"END_DEFINITION"</code>   <code>"START_WORKING"</code>   <code>"END_WORKING"</code>   <code>"CLEAR_ALL_OBJECT"</code>   <code>"CHANGE_DEFINITION"</code>   <code>"ADD_DEFINITION"</code> ]
<Identifier>	► [ <code>&lt;Letter&gt;</code>   <code>&lt;Underscore&gt;</code> ] [ <code>&lt;Letter&gt;</code>   <code>&lt;Decimaldigit&gt;</code>   <code>&lt;Underscore&gt;</code> ]*
<Millisec>	► [ <code>&lt;NonZeroDecimal&gt;</code> ][ <code>&lt;Decimaldigit&gt;*</code> ]
<Letter>	► <code>&lt;LowerCase&gt;</code>   <code>&lt;UpperCase&gt;</code>
<LowerCase>	► <code>"a"</code>   <code>."</code>   <code>"z"</code>
<UpperCase>	► <code>"A"</code>   <code>."</code>   <code>"Z"</code>
<Decimaldigit>	► <code>"0"</code>   <code>&lt;NonZeroDecimal&gt;</code>
<NonZeroDecimal>	► <code>"1"</code>   <code>"2"</code>   <code>"3"</code>   <code>"4"</code>   <code>"5"</code>   <code>"6"</code>   <code>"7"</code>   <code>"8"</code>   <code>"9"</code>
<Hexdigit>	► <code>"0"</code>   <code>"1"</code>   <code>"2"</code>   <code>"3"</code>   <code>"4"</code>   <code>"5"</code>   <code>"6"</code>   <code>"7"</code>   <code>"8"</code>   <code>"9"</code>   <code>"a"</code>   <code>"b"</code>   <code>"c"</code>   <code>"d"</code>   <code>"e"</code>   <code>"f"</code>   <code>"A"</code>   <code>"B"</code>   <code>"C"</code>   <code>"D"</code>   <code>"E"</code>   <code>"F"</code>
<Blank>	► [ <code>" "</code>   <code>"\t"</code> ] <sup>+</sup>
<Newline>	► <code>"\n"</code>

<ModulId>	► <Decimaldigit><Decimaldigit>*
<ModulName>	► <Identifier>
<InterfaceId>	► <Decimaldigit><Decimaldigit>*
<InterfaceName>	► <Identifier>
<OPName>	► <Identifier>
<COId>	► <Decimaldigit><Decimaldigit>*
<COName>	► <Identifier>
<TransitionVD>	► “VdHandle“<Blank> [<VdHandle> ”“”<VDName>”“”]
<VdHandle>	► <HexValue> <Decimaldigit>+

## 2.5 Makrodefinition (gesamt)

<MacroDefinition>

► START\_MACRO  
[GAT\_LOADDRIVER  
|GAT\_UNLOADDRIVER  
|GAT\_DELAY  
|GAT\_LABEL  
|GAT\_JUMP  
|GAT\_JUMP\_IF  
|GDI\_INITIATE  
|GDI\_CREATEFUNCOBJ  
|GDI\_CREATECOMMOBJ  
|GDI\_DELETEFUNCOBJ  
|GDI\_DELETECOMMOBJ  
|GDI\_READ  
|GDI\_WRITE  
|GDI\_EXECUTE  
|GDI\_CONCLUDE  
|GDI\_ABORT  
|GDI\_IDENT  
|GDI\_TRANSITION  
|GDI\_CREATEOPERATION  
|GDI\_DELETEOPERATION  
|GDI\_STATUS]\*  
STOP\_MACRO

## 2.6 Besonderheiten der Parametrierung

### 2.6.1 Datentyp Union

Der Datentyp Union (Syntax siehe <UnionValue>) wird unter [1] definiert. Es gelten die dort festgelegten Regeln.

Der Parameter „\_identifiziert1“ bezeichnet den Selektor, welcher den zu beschreibenden Zweig der Union festlegt. Im Makro ist demzufolge nur dieser Zweig zu beschreiben.

### 2.6.2 Datentyp String

Der Datentyp String (Syntax siehe <StringValue>) bezeichnet eine Zeichenkette. Längenangaben (wie beispielsweise in einer Sequenz) des Strings definieren die Anzahl der Zeichen sowie das Endezeichen oder Nullzeichen, welches den String abschließt.

### 2.6.3 Datentyp Sequenz

Der Datentyp Sequenz (Syntax siehe <SeqValue>) wird unter [1] definiert. Es gelten die dort festgelegten Regeln.

Der Parameter „Max“ definiert die reservierte und damit maximal zulässige Anzahl von eingetragenen Werten. Der Parameter „Length“ beschreibt die Anzahl der Elemente, welche aktuell in der Sequenz enthalten sind. Die Anzahl der Elemente der Sequenz bei der Parametrierung muß demzufolge gleich des in „Length“ eingetragenen Wertes sein.

### 2.6.4 Datentyp Enumeration

Der Datentyp Enumeration (Syntax siehe <EnumValue>) wird unter [1] definiert. Es gelten die dort festgelegten Regeln.

Für die Parametrierung sind die die Mitglieder repräsentierenden numerischen Werte aber auch die Bezeichnung der Mitglieder selbst zulässig.

### 2.6.5 Datentyp Funktionsreferenz (FRefValue)

Der Datentyp Funktionsreferenz (Syntax siehe <FRefValue>) definiert die Parameter einer vom GDI-Treiber generierten Funktionsobjekt-Instanz.

*Ürsprünglich werden dem GDI-Treiber vom Koordinator das API-Handle sowie die Funktions-Id aus der DCD als Funktionsreferenz übergeben. Da aber dem Makro das vom GDI-Treiber generierte zugehörige API-Handle nicht bekannt sein kann, wird dort anstelle des API-Handles der Name der Funktionsobjekt-Instanz notiert. Mit Hilfe dieses Instanznamens kann der Koordinator das API-Handle bilden.*

*Die seit Version 4.4 enthaltene Com-Objekt-Id ist reserviert für den Application Callback. Zur Spezifikation der Funktionsobjekt-Instanz hat diese Id aber keinerlei Bedeutung und wird deshalb nicht verwendet.*

### **2.6.6 VD-Handle (*VdHandle*)**

Das VD-Handle bezeichnet das vom GDI Treiber durch GDI\_Initiate generierte API-Handle, welches die erzeugte VD-Instanz beschreibt.

Es kann in der Regel aber nicht davon ausgegangen werden, dass der GDI-Treiber immer wieder das gleiche Handle für das entsprechende Modul vergibt.

### 3 Beispiel

Macro file generated by rd electronic gmbh, at Thu Apr 08 13:05:54 2010

```
[START_MACRO]
<VERSION> 1.1.0.0

[GAT_LOADDRIVER]
<DCDFILE> ..\DCD4_4\MDAQ_CS.dcd
<DRIVERFILE> ..\Run\MDAQDeviceDriver.dll
<DITLANGUAGE> en
<ALIGNMENT> 8
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll

[GDI_INITIATE]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<MODULENAME> vdMultiDataAcquisition
<VDID> 1400
<JOBID> 0
[BEGIN_INPUTPARAMETER] vdMultiDataAcquisition:InitiateParam.InitiateParamStruct
Max 9
Length 9
ConnectionParameters "USB:*:*"
hCorrespondingDeviceComponent 0
[END_INPUTPARAMETER]

[GDI_TRANSITION]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<TRANSITIONID> START_DEFINITION
<JOBID> 0

[GDI_IDENT]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<JOBID> 0

[GDI_STATUS]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<JOBID> 0

[GDI_CREATEFUNC OBJ]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<DEVICEFUNCNAME> fnCS_Inspection_1
<DEVICEFUNCID> 1020
<JOBID> 0
[NO_INPUTPARAMETER]

[GDI_CREATEFUNC OBJ]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<DEVICEFUNCNAME> fnCS_ControlCallBack_1
<DEVICEFUNCID> 1030
<JOBID> 0
[NO_INPUTPARAMETER]

[GDI_CREATEFUNC OBJ]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<DEVICEFUNCNAME> fnMDAQ_Device_1
<DEVICEFUNCID> 1440
<JOBID> 0
[NO_INPUTPARAMETER]

[GDI_TRANSITION]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<TRANSITIONID> END_DEFINITION
<JOBID> 0
```



```
[GDI_TRANSITION]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<TRANSITIONID> START_WORKING
<JOBID> 0
```

```
[GDI_TRANSITION]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<TRANSITIONID> END_WORKING
<JOBID> 0
```

```
[GDI_DELETEFUNCOBJ]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<DEVICEFUNCNAME> fnMDAQ_Device_1
<JOBID> 0
```

```
[GDI_TRANSITION]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<TRANSITIONID> CLEAR_ALL_OBJECT
<JOBID> 0
```

```
[GDI_CONCLUDE]
<DRIVERID> (1) [MDAQ_CS.dcd]C:\project\Run\MDAQDeviceDriver.dll
<VDNAME> vdMultiDataAcquisition_1
<JOBID> 0
```

```
[STOP_MACRO]
```