

VariablenII

Joachim Büchse

Copyright © CopyrightÂ©1993 Inovatronics, Inc.

COLLABORATORS

	<i>TITLE :</i> VariablenII		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Joachim BÜchse	February 14, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 VariablenII	1
1.1 Spezielle Variablen, Funktionen und Befehle	1
1.2 <Variable>=Alias({String})	2
1.3 Let {String}=BumpRevision({String})	2
1.4 Let {Ergebnis}=Case({Ganzzahl n}, {Ausdruck1} [,bis zu 10 Ausdrücke])	2
1.5 Let {HexString}=CharsToHex({String})	3
1.6 DeleteArrayEntry <ArrayVar>, {ArrayIndex}	3
1.7 Let {Ergebnis}=Evaluate({Ausdruck})	3
1.8 Let {Ergebnis}=EvaluateExpression({Ausdruck})	3
1.9 Let {ArrayIndex}=FirstArrayIndex(<ArrayVar>)	4
1.10 Let {String}=FormatValue({Wert}, {FormatString})	4
1.11 Let {RecordVar}=GetDBObject	4
1.12 Global <VarName>	4
1.13 Let {String}=HexToChars({HexString})	5
1.14 Let {Ganzzahl}=HexToInteger({HexString})	5
1.15 Let {Realzahl}=Infinity	5
1.16 Let {Ergebnis}=IfThen({Boolscher Wert}, {TrueAusdruck} [, {FalseAusdruck}])	5
1.17 Let {String}=Input({Prompt})	6
1.18 InsertArrayEntry <ArrayVar>, {ArrayIndex}	6
1.19 InsertRecordNames <RecordVar>	6
1.20 Let {HexString}=IntegerToHex({Ganzzahl})	6
1.21 Let {Boolscher Wert}=IsAlias(<VarName>)	7
1.22 Let {ArrayIndex}=LastArrayIndex(<ArrayVar>)	7
1.23 Let {Ganzzahl}=Limit({MinGrenze}, {MaxGrenze}, {Ganzzahl})	7
1.24 Let {Variable}=LoadVariable({DateiPfad})	7
1.25 Local <VarName>	8
1.26 Let {Ganzzahl}=Match({Ganzzahl oder String}, {TestDaten1} [,bis zu 10 TestDaten])	8
1.27 Let {ArrayIndex}=NextArrayIndex(<ArrayVar>, {ArrayIndex})	8
1.28 Let {Realzahl}=NotANumber	8
1.29 Let {Ganzzahl}=NumberOfArrayEntries(<ArrayVar>)	9

1.30	PatternMatch({String},{Muster})	9
1.31	Let {Ganzzahl}=PreviousArrayIndex(<ArrayVar>, {ArrayIndex})	9
1.32	SaveVariable <VarName>, {DateiPfad}	9
1.33	Let {ArrayIndex}=SearchArray(<ArrayVar>, {SuchDaten} [, {SchlüsselString} [, {StartIndex}]]))	10
1.34	SetDBObjects <RecordVar>	10
1.35	SetSearchArrayFlags <Flags>	10
1.36	SortArray <ArrayVar>, <Flags> [, {SchlüsselString}]	10
1.37	Static <VarName>	11
1.38	Let {Boolscher Wert}=ValidReal({Realzahl})	11
1.39	Let {String}=VarType(<VarName>)	11
1.40	Let {Boolscher Wert}=VerifyExpression({Ausdruck})	12

Chapter 1

VariablenII

1.1 Spezielle Variablen, Funktionen und Befehle

Funktion =Alias()

Funktion =BumpRevision()

Funktion =Case()

Funktion =CharsToHex()

Befehl DeleteArrayEntry

Funktion =Evaluate()

Funktion =EvaluateExpression()

Funktion =FirstArrayIndex()

Funktion =FormatValue()

Variable =GetDBObject

Befehl Global

Funktion =HexToChars()

Funktion =HexToInteger()

Funktion =IfThen()

Variable =Infinity

Funktion =Input()

Befehl InsertArrayEntry

Befehl InsertRecordNames

Funktion =IntegerToHex()

Funktion =IsAlias()

Funktion =LastArrayIndex()

Funktion =Limit()

Funktion =LoadVariable()

Befehl Local

Funktion =Match()

Funktion `=NextArrayIndex()`

Variable `=NotANumber`

Funktion `=NumberOfArrayEntries()`

Funktion `=PatternMatch()`

Funktion `=PreviousArrayIndex()`

Befehl `SaveVariable`

Funktion `=SearchArray()`

Befehl `SetDBObject`

Befehl `SetSearchArrayFlags`

Befehl `SortArray`

Befehl `Static`

Funktion `=ValidReal()`

Funktion `=VarType()`

Funktion `=VerifyExpression()`

1.2 `<Variable>=Alias({String})`

Befehl...

`<Variable>=Alias({String})`

Beschreibung...

Definiert die Variable als Alias für den, aus dem {String} erzeugten Variablennamen. Im Handbuch finden Sie eine genaue Funktionsbeschreibung.

Siehe auch: [IsAlias](#)

1.3 `Let {String}=BumpRevision({String})`

Befehl...

`Let {String}=BumpRevision({String})`

Beschreibung...

Erhöht die Versionsnummer des Strings in der selben Weise, wie die Diskcopy-Funktion der Workbench. Im Handbuch finden Sie eine genaue Funktionsbeschreibung.

1.4 `Let {Ergebnis}=Case({Ganzzahl n}, {Ausdruck1} [,bis zu 10 Ausdrücke])`

Befehl...

`Let {Ergebnis}=Case({Ganzzahl n}, {Ausdruck1} [,bis zu 10 Ausdrücke])`

Beschreibung...

Gibt das Ergebnis der Berechnung des n-ten Ausdrucks zurück.

Zum Beispiel:

Let X = 2

Echo Case(X,"Gold","Silber","Bronze")

Ausgabe: Silber

Siehe auch: [IfThen Match](#)

1.5 Let {HexString}=CharsToHex({String})

Befehl...

Let {HexString}=CharsToHex({String})

Beschreibung...

Wandelt einen normalen String in einen doppelt so langen

Hex-String.

Echo CharsToHex("CanDo 2.0")

Ausgabe: 43616E446F2056322E30

Siehe auch: [HexToChars](#) [FileWriteChars](#)

1.6 DeleteArrayEntry <ArrayVar>, {ArrayIndex}

Befehl...

DeleteArrayEntry <ArrayVar>, {ArrayIndex }

Beschreibung...

Löscht den Array-Eintrag mit dem angegebenen Index aus dem

Array. Alle darüberliegenden Einträge werden, beginnend mit

{ArrayIndex}, neu durchnummeriert.

Siehe auch: [InsertArrayEntry](#) [Dispose](#)

1.7 Let {Ergebnis}=Evaluate({Ausdruck})

Befehl...

Let {Ergebnis}=Evaluate({Ausdruck})

Siehe: [EvaluateExpression](#)

1.8 Let {Ergebnis}=EvaluateExpression({Ausdruck})

Befehl...

Let {Ergebnis}=EvaluateExpression({Ausdruck})

Beschreibung...

Wenn der String einen gültigen Ausdruck enthält, so wird das

Ergebnis dessen Berechnung zurückgegeben. Ist der Ausdruck

ungültig, so erzeugt CanDo einen Fehler. Benutzen Sie

[VerifyExpression](#) um dieses Problem auszuschalten.

1.9 Let {ArrayIndex}=FirstArrayIndex(<ArrayVar>)

Befehl...

```
Let {ArrayIndex}=FirstArrayIndex(<ArrayVar>)
```

Beschreibung...

Gibt den kleinsten in diesem Array existierenden Index zurück.

Wenn das Array keine Einträge besitzt, so wird der Wert 0 zurückgegeben und SearchFound auf False gesetzt.

Siehe auch: [LastArrayIndex](#)

1.10 Let {String}=FormatValue({Wert}, {FormatString})

Befehl...

```
Let {String}=FormatValue({Wert}, {FormatString})
```

Beschreibung...

Formatiert den angegebenen Zahlenwert entsprechend der Definition, die durch den {FormatString} vorgegeben wird. Im Handbuch finden Sie die genaue Funktionsbeschreibung.

1.11 Let {RecordVar}=GetDBObjects

Befehl...

```
Let {RecordVar}=GetDBObjects
```

Beschreibung...

Gibt eine Record-Variable zurück, die die Inhalte aller aktuell auf der Card befindlichen Datenbank-Objekte enthält. Genaue Funktionsbeschreibung siehe Handbuch.

Siehe auch: [SetDBObjects](#)

1.12 Global <VarName>

Befehl...

```
Global <VarName>
```

Beschreibung...

Alle Variablen die nicht mittels [Local](#) oder [Static](#) definiert wurden, sind global. Dieser Befehl wurde nur der Vollständigkeit halber implementiert.

1.13 Let {String}=HexToChars({HexString})

Befehl...

```
Let {String}=HexToChars({HexString})
```

Beschreibung...

Der angegebene Hex-String wird in einen normalen String gewandelt. Ein Hex-String wird zum Beispiel vom Befehl `FileReadChars` zurückgegeben.

Siehe auch: [CharsToHex](#)

1.14 Let {Ganzzahl}=HexToInteger({HexString})

Befehl...

```
Let {Ganzzahl}=HexToInteger({HexString})
```

Beschreibung...

Der angegebene Hex-String wird in eine Ganzzahl gewandelt. Ein Hex-String wird zum Beispiel vom Befehl `FileReadChars` zurückgegeben.

Siehe auch: [IntegerToHex](#)

1.15 Let {Realzahl}=Infinity

Befehl...

```
Let {Realzahl}=Infinity
```

Beschreibung...

Gibt den IEEE-Fließkommawert für Unendlich zurück. Im Handbuch finden Sie eine genaue Beschreibung dieser Systemvariablen.

Siehe auch: [NotANumber](#) [ValidReal](#)

1.16 Let {Ergebnis}=IfThen({Boolscher Wert}, {TrueAusdruck} [, {FalseAusdruck}])

Befehl...

```
Let {Ergebnis}=IfThen({Boolscher Wert}, {TrueAusdruck} [, {FalseAusdruck}])
```

Beschreibung...

Gibt den `{TrueAusdruck}` zurück wenn der `{Boolsche Wert}` `True` ist. Sonst wird der `{FalseAusdruck}` oder, falls dieser nicht angegeben wurde, `Nothing` zurückgegeben.

Siehe auch: [Match Case](#)

1.17 Let {String}=Input({Prompt})

Befehl...

```
Let {String}=Input({Prompt})
```

Beschreibung...

Zeigt den "Prompt" im CLI an, um eine Eingabezeile vom Benutzer zu erfragen. Es muß ein CLI geöffnet sein, um mit dieser Funktion zu arbeiten. Die Eingabe wird mit der Returnntaste abgeschlossen.

1.18 InsertArrayEntry <ArrayVar>, {ArrayIndex}

Befehl...

```
InsertArrayEntry <ArrayVar>, {ArrayIndex}
```

Beschreibung...

Erzeugt Platz für einen neuen Eintrag mit dem angegebenen Index im Array. Alle darüberliegenden Einträge werden beginnend mit {ArrayIndex}+1 neu durchnummeriert.

Siehe auch: [DeleteArrayEntry](#)

1.19 InsertRecordNames <RecordVar>

Befehl...

```
InsertRecordNames <RecordVar>
```

Beschreibung...

Schreibt die Namen aller Einträge in der angegebenen Record-Variablen ins aktuelle Dokument. Dieser Befehl kann verwendet werden, um die Namen von unbekanntem Einträgen zu ermitteln.

1.20 Let {HexString}=IntegerToHex({Ganzzahl})

Befehl...

```
Let {HexString}=IntegerToHex({Ganzzahl})
```

Beschreibung...

Wandelt die {Ganzzahl} in einen Hex-String. Dieser String kann zwei, vier, sechs oder acht Zeichen lang sein.

```
Echo IntegerToHex(1256)
```

Ausgabe: 04E8

Siehe auch: [HexToInteger](#) [CharsToHex](#)

1.21 Let {Boolscher Wert}=IsAlias(<VarName>)

Befehl...

Let {Boolscher Wert}=IsAlias(<VarName>)

Beschreibung...

Gibt den Wert True zurück, wenn die angegebene Variable ein Alias ist.

Siehe auch: [Alias](#)

1.22 Let {ArrayIndex}=LastArrayIndex(<ArrayVar>)

Befehl...

Let {ArrayIndex}=LastArrayIndex(<ArrayVar>)

Beschreibung...

Gibt den größten in diesem Array existierenden Index zurück.

Wenn das Array keine Einträge besitzt, wird der Wert 0 zurückgegeben und SearchFound auf False gesetzt.

Siehe auch: [FirstArrayIndex](#)

1.23 Let {Ganzzahl}=Limit({MinGrenze}, {MaxGrenze}, {Ganzzahl})

Befehl...

Let {Ganzzahl}=Limit({MinGrenze}, {MaxGrenze}, {Ganzzahl})

Beschreibung...

Gibt die {Ganzzahl} zurück, wenn diese zwischen den Zahlen {MinGrenze} und {MaxGrenze} liegt. Ist sie größer als beide Grenzen, so wird die größere der beiden Grenzen zurückgegeben; Ist sie kleiner als beide Grenzen, wird die kleinere Grenze zurückgegeben.

1.24 Let {Variable}=LoadVariable({DateiPfad})

Befehl...

Let {Variable}=LoadVariable({DateiPfad})

Beschreibung...

Gibt den Inhalt der gespeicherten Variablen zurück.

Siehe auch: [SaveVariable](#)

1.25 Local <VarName>

Befehl...

Local <VarName>

Beschreibung...

Der Befehl definiert, daß die Variable nur vom aktuellen Script aus benutzt werden kann und nach dessen Ausführung gelöscht wird. Eine globale Variable gleichen Namens wird dadurch verschattet (unsichtbar).

Siehe auch: [Static Global](#)

1.26 Let {Ganzzahl}=Match({Ganzzahl oder String}, {TestDaten1} [,bis zu 10 Test-Daten])

Befehl...

Let {Ganzzahl}=Match({Ganzzahl oder String}, {TestDaten1} [,bis zu 10 TestDaten])

Beschreibung...

Gibt die Nummer der Testdaten zurück, die mit der Ganzzahl bzw. dem String übereinstimmen, oder 0 falls keine übereinstimmen.

Genaue Beschreibung siehe Handbuch.

Siehe auch: [Case IfThen](#)

1.27 Let {ArrayIndex}=NextArrayIndex(<ArrayVar>, {ArrayIndex})

Befehl...

Let {ArrayIndex}=NextArrayIndex(<ArrayVar>, {ArrayIndex})

Beschreibung...

Gibt den nächstgrößeren, im Array existierenden Index nach {ArrayIndex} an. Existiert kein größerer Index, so wird der übergebene {ArrayIndex} zurückgegeben und die Systemvariable SearchFound auf False gesetzt.

Siehe auch: [PreviousArrayIndex](#)

1.28 Let {Realzahl}=NotANumber

Befehl...

Let {Realzahl}=NotANumber

Beschreibung...

Dieser Wert entspricht dem Ergebnis einiger Fließkomma-Berechnungen, die kein numerisches Ergebnis liefern. Genaue Beschreibung dieser Konstanten siehe Handbuch.

Siehe auch: [Infinity ValidReal](#)

1.29 Let {Ganzzahl}=NumberOfArrayEntries(<ArrayVar>)

Befehl...

Let {Ganzzahl}=NumberOfArrayEntries(<ArrayVar>)

Beschreibung...

Gibt die Anzahl der Einträge im angegebenen Array zurück.

1.30 PatternMatch({String},{Muster})

Befehl...

Let {Boolsche Variable} = PatternMatch({String},{Muster})

Beschreibung...

Diese Funktion testet, ob der String dem DOS-Muster entspricht.

Ist dies der Fall, so gibt sie TRUE zurück, sonst FALSE. Eine

Beschreibung der DOS-Muster finden Sie in Ihrem AmigaDOS- oder

Workbench-Handbuch.

1.31 Let {Ganzzahl}=PreviousArrayIndex(<ArrayVar>, {ArrayIndex})

Befehl...

Let {Ganzzahl}=PreviousArrayIndex(<ArrayVar>, {ArrayIndex})

Beschreibung...

Gibt den nächstkleineren, im Array existierenden Index vor

{ArrayIndex} an. Existiert kein kleinerer Index, so wird der

übergebene {ArrayIndex} zurückgegeben und die Systemvariable

SearchFound auf False gesetzt.

Siehe auch: [NextArrayIndex](#)

1.32 SaveVariable <VarName>, {DateiPfad}

Befehl...

SaveVariable <VarName>, {DateiPfad}

Beschreibung...

Speichert den Inhalt der Variablen unter dem angegebenen Pfad

ab. Wenn die Variable ein Array oder Record ist, so wird der

gesamte Datensatz abgespeichert.

Siehe auch: [LoadVariable](#)

1.33 **Let {ArrayIndex}=SearchArray(<ArrayVar>, {SuchDaten} [, {SchlüsselString} [, {StartIndex}]]])**

Befehl...

Let {ArrayIndex}=SearchArray(<ArrayVar>, {SuchDaten} [, {SchlüsselString} [, {StartIndex}]]])

Beschreibung...

Sucht die angegebenen Daten im Array. Mit dem {SchlüsselString} kann die Suche auf einen Record-Eintrag oder ein Sub-Array beschränkt werden.

Siehe auch: [SetSearchArrayFlags](#) [SearchFound](#)

1.34 **SetDBObjects <RecordVar>**

Befehl...

SetDBObjects <RecordVar>

Beschreibung...

Setzt alle Datenbank-Objekte der aktuellen Card auf die Werte der gleichnamigen Einträge im angegeben Record.

Siehe auch: [GetDBObjects](#)

1.35 **SetSearchArrayFlags <Flags>**

Befehl...

SetSearchArrayFlags <Flags>

Beschreibung...

Setzt die Suchflags für die Funktion [SearchArray](#) .

Im Handbuch finden Sie eine genaue Beschreibung der Flags.

Flags:

FORWARD, BACKWARD,

NOCASE,

SUBSTRING, SUPERSTRING

1.36 **SortArray <ArrayVar>, <Flags> [, {SchlüsselString}]**

Befehl...

SortArray <ArrayVar>, <Flags> [, {SchlüsselString}]

Beschreibung...

Sortiert die Einträge im angegebenen Array. Genaue Funktionsbeschreibung siehe Handbuch.

1.37 Static <VarName>

Befehl...

Static <VarName>

Beschreibung...

Entspricht dem Befehl **Local** mit dem Unterschied, daß die Variablen beim nächsten Aufruf des Scripts noch die alten Werte enthalten.

Siehe auch: **Global**

1.38 Let {Boolscher Wert}=ValidReal({Realzahl})

Befehl...

Let {Boolscher Wert}=ValidReal({Realzahl})

Beschreibung...

Gibt True zurück, falls es sich bei der angegebenen Realzahl um eine existierende, nicht unendliche Zahl handelt. Genaue Funktionsbeschreibung siehe Handbuch.

Siehe auch: **NotANumber Infinity**

1.39 Let {String}=VarType(<VarName>)

Befehl...

Let {String}=VarType(<VarName>)

Beschreibung...

Gibt den Typ der Variablen zurück:

String

Integer = Ganzzahl

Real = Fließkommazahl

Logical = Boolesch

Nothing = Nichts

System

Array

Record

Alias Variablen können nur mit Hilfe der der Funktion

IsAlias erkannt werden.

1.40 Let {Boolscher Wert}=VerifyExpression({Ausdruck})

Befehl...

Let {Boolscher Wert}=VerifyExpression({Ausdruck})

Beschreibung...

Gibt True zurück wenn der {Ausdruck} gültig ist, sonst

False .

Der Befehl wird meist vor der Funktion **EvaluateExpression**

benutzt, um Fehler zu vermeiden.
