



# ESS

## Enterprise Solution Server



## Dokumentation

# 'Programmierhandbuch'



# Inhaltsangabe

<b>1 Die Programmiersprachen.....</b>	<b>8</b>
<b>2 Masken.....</b>	<b>9</b>
<b>2.1 Beschreibung der Syntax.....</b>	<b>9</b>
2.1.1 Module.....	9
2.1.2 Masken und Screens.....	9
2.1.3 Frames.....	10
2.1.4 Felder.....	11
2.1.5 Formscripten.....	14
<b>2.2 Beschreibung der Sprachelemente.....</b>	<b>16</b>
2.2.1 Module.....	16
OS_MODULE.....	16
2.2.2 Masken und Screens.....	17
2.2.2.1 Maskentypen <masktype>.....	17
OS_HOMEPAGE.....	17
OS_MASK.....	17
OS_MENU.....	17
OS_MENUBAR.....	17
OS_SPECMASK.....	17
2.2.2.2 Maskenauszeichnungen <maskst>.....	18
OS_HIDDEN.....	18
OS_NOWARN.....	18
OS_SELUID.....	18
OS_SYSPERM.....	18
OS_TITLE.....	18
OS_POSMASK.....	19
OS_MMENUBAR.....	19
OS_IMEXMENUBAR.....	19
OS_ORDERMENUBAR.....	19
OS_SELECTMENUBAR.....	19
OS_SURVOMENUBAR.....	20
OS_SURVEYMENUBAR.....	20
2.2.2.3 Maskenänderungsanweisungen <updatemaskst>.....	20
OS_SORTLIST.....	20
OS_INDEXLIST.....	20
OS_NMGIVELIST.....	21
OS_NMTAKELIST.....	21
OS_MFIELDLIST.....	22
2.2.2.4 Listenpositionen in Masken <listpos>.....	22
OS_BEFORE.....	22
OS_AFTER.....	23
OS_FIRST.....	23
OS_LAST.....	23



OS_NOMPD.....	23
2.2.3 Frames.....	24
2.2.3.1 Framedefinition.....	24
OS_FRAME.....	24
2.2.3.2 Frameauszeichnungen <framest>.....	24
OS_BORDERCOLOR.....	24
OS_BORDERWIDTH.....	24
OS_BORDERMAP.....	24
OS_BORDERTYPE.....	25
OS_TLCOLOR.....	25
OS_BGCOLOR.....	25
OS_LINEHEIGHT.....	25
OS_TITLE.....	25
OS_TITLEPOS.....	25
OS_TALIGN.....	26
OS_MFIELDLIST.....	26
2.2.3.3 Frameänderungsanweisungen <updateframe>.....	27
OS_UPDATEFRAME.....	27
2.2.4 Felder.....	28
2.2.4.1 Felddefinition.....	28
2.2.4.2 Feldeigenschaften <fieldlist>.....	28
OS_ACTION.....	28
OS_AICMP.....	30
OS_CHECKDEFAULT.....	30
OS_COMBI.....	30
OS_CURCOMBI.....	30
OS_DBINFO.....	30
OS_DEFAULT.....	31
OS_EMPTY.....	31
OS_FKEY.....	31
OS_FORCEDEFAULT.....	31
OS_IMAGE.....	31
OS_ISSURVEY.....	31
OS_JOINFIELD.....	32
OS_NOCHECK.....	32
OS_NOFILL.....	32
OS_NOPERMS.....	32
OS_SELDEFAULT.....	32
OS_SELF.....	33
OS_TITLE.....	33
2.2.4.3 Datenbankflags <dbtypeflag>.....	33
OS_ALPHACOUNT.....	33
OS_GROUP.....	33
OS_NUMCOUNT.....	33
OS_SERIAL.....	33
OS_UNIQUEDEL.....	33
OS_VARCHARTEXT.....	34
2.2.4.4 Feldeigenschaftsänderungen <updatefieldst>.....	34



OS_ACCELL.....	34
OS_ACCESS.....	34
OS_BGCOLOR.....	34
OS_DEFLIST.....	34
OS_DENYNMPD.....	35
OS_FALIGN.....	35
OS_FGCOLOR.....	35
OS_FORCE.....	35
OS_GIVEFIELDLIST.....	35
OS_NEXTMASK.....	36
OS_HEIGHT.....	36
OS_LINEHEIGHT.....	36
OS_NOFOCUS.....	36
OS_NOBORDER.....	36
OS_NOPOSCHECK.....	36
OS_ONELINE.....	36
OS_REGEXPR.....	37
OS_REFERENCELIST.....	37
OS_TAKEFIELDLIST.....	37
OS_TITLELEN.....	37
OS_TALIGN.....	37
OS_TLBGCOLOR.....	38
OS_TLCOLOR.....	38
OS_TITLEPOS.....	38
OS_TITLSTRING.....	38
OS_VISLEN.....	38
2.2.4.5 Feldfüller <fillflags>.....	39
OS_FILLACTCUR.....	39
OS_FILLACTDATE.....	39
OS_FILLACTKW.....	39
OS_FILLACTTIME.....	39
OS_FILLLANG.....	39
OS_FILLMASKNUM.....	39
OS_FILLTABNAME.....	39
OS_FILLUID.....	40
OS_FILLTELNO.....	40
2.2.4.6 Selektionen <tgflag>.....	40
OS_NULL.....	40
OS_TGSEL.....	40
OS_USERSEL.....	40
2.2.4.7 Feldtypen <fieldtype>.....	41
OS_AIPD.....	41
OS_AREACODE.....	41
OS_BARPD.....	41
OS_BOOL.....	41
OS_BUTTON.....	41
OS_COUNTRYSIGN.....	41
OS_CURFACT.....	41
OS_CURRENCY.....	41
OS_DATE.....	41



OS_DIAGPD.....	41
OS_DOUBLE.....	42
OS_ENUMPD.....	42
OS_ENUMRB.....	42
OS_FAXNUMBER.....	42
OS_GIF.....	42
OS_HEX.....	42
OS_INT.....	42
OS_LABEL.....	42
OS_MAIL.....	42
OS_MASKPDBUTTON.....	42
OS_FORMPDBUTTON.....	42
OS_SBOOL.....	42
OS_SDATE.....	43
OS_SHORT.....	43
OS_STATUS.....	43
OS_STEXT.....	43
OS_SUBMASK.....	43
OS_TELNO.....	43
OS_TEXT.....	43
OS_TEXTAREA.....	43
OS_THUMBNAIL.....	43
OS_TIME.....	43
OS_UID.....	43
OS_WWW.....	43
2.2.4.8 Feldberechtigungen <accessval>.....	44
OS_READ.....	44
OS_WRITE.....	44
OS_EXEC.....	44
OS_NOACCESS.....	44
2.2.4.9 Titelpositionen <titlepos>.....	44
OS_PLEFT.....	44
OS_PRIGHT.....	44
OS_PABOVE.....	45
OS_PBELOW.....	45
2.2.5 Formscripten.....	46
2.2.5.1 Neue Skripteinträge.....	46
OS_CREATEFORMSCRIPT.....	46
2.2.5.2 Skripteinträge ändern.....	46
OS_UPDATEFORMSCRIPT.....	46
<b>3 Formulare.....</b>	<b>47</b>
<b>3.1 Beschreibung der Syntax.....</b>	<b>47</b>
<b>3.2 Beschreibung der Sprachelemente.....</b>	<b>49</b>
3.2.1 Funktionen.....	49
3.2.1.1 Funktionsdefinition.....	49
3.2.1.2 Besondere Funktionsdefinitionen.....	49



3.2.2 Vordefinierte Funktionen.....	49
setcurfieldcont.....	49
setcurfielderror.....	50
setcurfieldaccess.....	50
getcurfieldcont.....	50
setcurmaskcont.....	51
getcurmaskcont.....	51
getmasktitle.....	51
sepstrcpy.....	52
cellprops.....	52
makecell.....	53
newpage.....	53
date2langdate.....	54
langdatetodate.....	54
flangstring.....	54
dateadd.....	55
currency2intern.....	55
intern2currency.....	55
printcurrency.....	56
uxtime.....	56
uxtimes.....	57
itoa.....	57
atoi.....	57
ftoa.....	57
atof.....	58
round.....	58
mround.....	58
lineprops.....	59
pagesize.....	59
orientation.....	60
openrw.....	60
closerw.....	61
fdprocopen.....	61
fdprocclose.....	62
readline.....	62
writeline.....	63
printdocument.....	63
strlen.....	64
tempnam.....	64
strcmp.....	64
strncmp.....	65
getmxuid.....	65
getmxlogname.....	65
getmxhome.....	66
getflogo.....	66
getnlogo.....	66
ferrsprintf.....	67
fnotesprintf.....	67
3.2.3 Vordefinierte Datenbankfunktionen.....	67
dbreadlock.....	67



dbopencursor.....	68
dbfetchcursor.....	68
dbclosecursor.....	69
dbdelete.....	69
dbinsert.....	69
dbupdate.....	70
<b>3.3 Variablen.....</b>	<b>70</b>
3.3.1 Variablentypen.....	70
Variablentyp int.....	70
Variablentyp float.....	70
Variablentyp text.....	70
Variablentyp mask.....	71
3.3.2 Vordefinierte Variablen.....	71
curcell.....	71
globalvector.....	71
int curpage.....	71
int firstpageheader.....	72
int lastpagefooter.....	72
int date.....	72
int time.....	72
int pcurprec.....	72
int errno.....	72
<b>4 Perl-Commandos.....</b>	<b>73</b>
<b>4.1 Beschreibung der Funktionen.....</b>	<b>73</b>



# 1 Die Programmiersprachen

Die Programmiersprachen des ESS sind in zwei Bereiche zu teilen

- Beschreibungssprache für Masken und Felder
- Prozedurale Sprache mit vielen Elementen aus der Programmiersprache `C` für Formulare.

Als Felder werden im ESS alle Oberflächenelemente bezeichnet, die über Tastatur oder Maus bedienbar sind.

Masken sind eine Zusammenfassung von verschiedenen Feldern zu einer Objekt.

Alle Positionierungsangaben für Felder und Frames sind in Pixel-Angaben.

Formulare sind ausführbare Programme, die z.B. Dokumente erzeugen oder Berechnungen auf den Masken ausführen.

Alle Grössen- und Positionierungsangaben sind in 10-tel Millimeter-Angaben.





## 2 Masken

### 2.1 Beschreibung der Syntax

#### 2.1.1 Module

<modulelist>	:=	<module>
		<modulelist>
	;	
<module>	:=	OS_MODULE ident <number> <number>
		'{' <maskorfieldlist >'}
	;	
<maskorfieldlist>	:=	<maskorfield>
		<maskorfield> <maskorfieldlist>
	;	
<maskorfield>	:=	<mask>
		<field>
		<frame>
		<createscriptst>
		<updatemask>
		<updatefield>
		<updateframe>
		<updateformscriptst>
	;	

#### 2.1.2 Masken und Screens

<mask>	:=	<masktype> ident '{' <maskstlist> '}'
	;	
<maskstlist>	:=	<maskst>
		<maskstlist>
	;	
<masktype>	:=	OS_HOMEPAGE
		OS_MASK
		OS_MENU
		OS_MENUBAR
		OS_SPECMASK
	;	
<maskst>	:=	<updatemaskst>
		OS_HIDDEN
		OS_NOWARN
		OS_SELUID ';'
		OS_SYSPERM ';'
		OS_TITLE <number> ';'
		OS_POSMASK ident ';'
		OS_MMENUBAR ident ';'
		OS_IMEXMENUBAR ident ';'
		OS_ORDERMENUBAR ident ';'
		OS_SELECTMENUBAR ident ';'
		OS_SURVOMENUBAR ident ';'
		OS_SURVEYMENUBAR ident ';'



	:=	' ;'
	;	
<updatemaskst>	:=	OS_SORTLIST '{' '{' <orderlist> '}' '}'
		OS_INDEXLIST [ OS_UNIQUE ] '{' <orderlist> '}'
		OS_NMGIVELIST [ OS_REVERSE ] <listpos>
		'{' <nmgivelist> '}'
		OS_NMTAKELIST [ OS_REVERSE ] '{' <nmgivelist> '}'
		OS_MFIELDLIST '{' <maskfieldlist> '}'
		' ;'
	;	
<orderlist>	:=	ident ',' <number> ' ;'
	;	
<listpos>	:=	<listposition> <listtarget>
	;	
<listposition>	:=	OS_BEFORE
		OS_AFTER
	;	
<listtarget>	:=	OS_FIRST
		OS_LAST
		ident
	;	
<nmgivelist>	:=	ident [ OS_NOMPD ] '{' <takefieldlist> '}'
		OS_SEPARATOR ident ' ;'
	;	
<takefieldlist>	:=	ident ',' ident ' ;'
		ident ',' ident ' ' <tgflaglist> ' ;'
		<takefieldlist>
	;	
<tgflaglist>	:=	<tgflag>
		<tgflaglist>
	;	
<tgflag>	:=	OS_NULL
		OS_TGSEL
		OS_USERSEL
		OS_COPY
	;	
<maskfieldlist>	:=	<addormove> OS_SCREEN <number> OS_TITLE <number>
		'{' framelist '}'
		OS_DELETE OS_SCREEN <number> ' ;'
	;	
<framelist>	:=	<addormove> ident '[' <number> ' ' <number> ']' ' ;'
		OS_UPDATE ident ' ;'
	;	
<addormove>	:=	OS_ADD
		OS_MOVE
	;	

### 2.1.3 Frames

<frame>	:=	OS_FRAME ident '[' <number> ' ' <number> ']'
		'{' <framestlist> '}'



	;	
<framestlist>	:=	<framest> <framestlist>
	;	
<framest>	:=	OS_BORDERCOLOR <number> ','
		OS_BORDERWIDTH <number> ','
		OS_BORDERMAP <number> ','
		OS_BORDERTYPE <number> ','
		OS_TLCOLOR <number> ','
		OS_BGCOLOR <number> ','
		OS_LINEHEIGHT <number> ','
		OS_TITLE <number> ','
		OS_TITLEPOS <number> ','
		OS_TALIGN <number> ','
		OS_MFIELDLIST '{' <positionfieldlist> '}'
	;	
<updateframe>	:=	OS_UPDATEFRAME ident '[' <number> ',' <number> ']'
		'{' <framestlist> '}'
	;	
<positionfieldlist>	:=	<addormove> ident '[' <number> ',' <number> ']' ','
		OS_DELETE ident ','
<addormove>	:=	OS_ADD
		OS_MOVE
	;	

## 2.1.4 Felder

<field>	:=	<fieldtype> ident <sizes> <accesslist> '{' <fieldstlist> '}'
	;	
<fieldstlist>	:=	<fieldst> <fieldstlist>
	;	
<fieldst>	:=	<updatefieldst>
		OS_ACTION ident ','
		OS_AICMP ','
		OS_CHECKDEFAULT ','
		OS_COMBI ident ','
		OS_CURCOMBI ident ','
		OS_DBINFO ident ',' ident ','
		OS_DEFAULT <defaultvalue> ','
		OS_EMPTY ','
		OS_FKEY <decnum> ','
		OS_FORCEDEFAULT <defaultvalue> ','
		OS_IMAGE string ','
		OS_ISSURVEY '(' <number> ',' <number> ')' ','
		OS_JOINFIELD ident ','
		OS_NOCHECK ','
		OS_NOFILL ','
		OS_NOPERMS ','
		OS_SELDEFAULT <defaultvalue> ','
		OS_SELF ','
		OS_TITLE <number> OS_TITLEPOS <titlepos> ','
		<dbtypeflag> ','



	;	
<updatefieldst>	:=	OS_ACCELL string ';' ; OS_ACCESS <accesslist> ';' ; OS_BGCOLOR <number> ';' ; OS_COLUMNS <number> ';' ; OS_DEFLIST <listpos> ' (' <deflist> ')' ';' ; OS_DEFLIST OS_MASKNAME <listpos> ' (' <maskdeflist> ')' ';' ; OS_DENYNMPD ';' ; OS_FALIGN <number> ';' ; OS_FGCOLOR <number> ';' ; OS_FORCE ';' ; OS_GIVEFIELDLIST '{' <takefieldlist> '}' ; OS_HEIGHT <number> ';' ; OS_LINEHEIGHT <number> ';' ; OS_NEXTMASK ident ';' ; OS_NOFOCUS ';' ; OS_NOBORDER ';' ; OS_NOPOSCHECK ';' ; OS_ONELINE ';' ; OS_REFERENCELIST '{' <locationlist> '}' ; OS_REGEXPR string ';' ; OS_TAKEFIELDLIST '{' <takefieldlist> '}' ; OS_TALIGN <number> ';' ; OS_TLBGCOLOR <number> ';' ; OS_TLCOLOR <number> ';' ; OS_TITLEPOS <titlepos> ';' ; OS_TITLSTRING ';' ; OS_TITLELEN <number> ';' ; OS_TITLEHEIGHT <number> ';' ; OS_TITLEFONT string ';' ; OS_TITLEFONTTYPE <number> ';' ; OS_FONT string ';' ; OS_FONTTYPE <number> ';' ; OS_VISLEN <number> ';' ;
	;	
<fieldtype>	:=	OS_AIPD OS_AREACODE OS_BARPD OS_BOOL OS_BUTTON OS_COUNTRYSIGN OS_CURFACT OS_CURRENCY OS_DATE OS_DIAGPD OS_DOUBLE OS_ENUMPD OS_ENUMRB OS_FAXNUMBER OS_GIF



		OS_HEX
		OS_INT
		OS_LABEL
		OS_MAIL
		OS_MASKPDBUTTON
		OS_FORMPDBUTTON
		OS_SBOOL
		OS_SDATE
		OS_SHORT
		OS_STATUS
		OS_STEXT
		OS_SUBMASK
		OS_TELNO
		OS_TEXT
		OS_TEXTAREA
		OS_THUMBNAIL
		OS_TIME
		OS_UID
		OS_WWW
		;
<fillflags>	:=	OS_FILLACTCUR
		OS_FILLACTDATE
		OS_FILLACTKW
		OS_FILLACTTIME
		OS_FILLDEFCUR
		OS_FILLDOCUDIR
		OS_FILLCOUNT
		OS_FILLDEBITOR
		OS_FILLKREDITOR
		OS_FILLJAVATIME
		OS_FILLLANG
		OS_FILLMASKNUM
		OS_FILLTABNAME
		OS_FILLUID
		OS_FILLTELNO
		<string>
		;
<dbtypeflag>	:=	OS_ALPHACOUNT ;'
		OS_GROUP ;'
		OS_NUMCOUNT ;'
		OS_SERIAL ;'
		OS_UNIQUEDEL ;'
		OS_VARCHARTEXT ;'
		;
<listpos>	:=	<listposition> <listtarget>
		;
<listposition>	:=	OS_BEFORE
		OS_AFTER
		;
<listtarget>	:=	OS_FIRST
		OS_LAST



		ident
	;	
<locationlist>	:=	ident ',' ident ';' <locationlist>
	;	
<takefieldlist>	:=	ident ',' ident ';' <takefieldlist>
		ident ',' ident ';' <tgflaglist> ';'
		<takefieldlist>
	;	
<tgflaglist>	:=	<tgflag>
		<tgflag> ' ' <tgflaglist>
	;	
<tgflag>	:=	OS_NULL
		OS_TGSEL
		OS_USERSEL
		OS_COPY
	;	
<sizes>	:=	'[' <number> ',' <number> ']'
	;	
<deflist>	:=	<number> ',' <deflist>
		<number>
	;	
<maskdeflist>	:=	ident ',' <maskdeflist>
		ident
	;	
<accesslist>	:=	<accessval> ' ' <accesslist>
		<accessval>
	;	
<accessval>	:=	OS_READ
		OS_WRITE
		OS_EXEC
		OS_NOACCESS
	;	
<titlepos>	:=	OS_PLEFT
		OS_PRIGHT
		OS_PABOVE
		OS_PBELOW
	;	
<number>	:=	Decnum
		Octnum
		Hexnum
	;	

### 2.1.5 Formscripten

<createscriptst>	:=	OS_CREATEFORMSCRIPT ident <numorsep> <repchangent>
	;	','
	;	
<updateformscriptst>	:=	OS_UPDATEFORMSCRIPT ident <scripttype> <formtype>
	;	ident <optprinter> <repchangent> ';'
	;	
<scripttype>	:=	OS_UPDATE



		OS_INSERT
		OS_DELETE
		OS_MASK
		OS_INPUTACTION
		OS_FIELD ident
		OS_FORMPULLDOWN
		;
<formtype>	:=	OS_PRINT
		OS_SURVPRINT
		;
<optprinter>	:=	OS_PRINTER ident
		;
<numorsep>	:=	<separatorst>
		<number>
		;
<separatorst>	:=	OS_SEPARATOR ident ';' ;
		;
<repchangest>	:=	OS_REPLACE ident ';' ;
		<changest>
		;
<changest>	:=	OS_BEFORE <placest> ';' ;
		OS_AFTER <placest> ';' ;
		;
<placest>	:=	<number>
		ident
		OS_FIRST
		OS_LAST
		;



## 2.2 Beschreibung der Sprachelemente

### 2.2.1 Module

#### ➤ OS\_MODULE

Definition .:

```
OS_MODULE ident <number> <number> {  
    ...  
}
```

Beschreibung:

Einleitung einer Maskenbeschreibung oder Masken- bzw. Feldänderung. Aufgrund der vergebenen Modul- und Entwicklernummer ist das entwickelte Modul Lizenzierungspflichtig, bzw. bei Verwendung von Modulnummer '0' und Entwicklernummer '0' wird die normale ESS-Lizenz verwendet.

Parameter:

- ident - Zeichenkette, Name des Modules
- number - Ganzzahl, Modulnummer. Die dazugehörigen Nummernkreise werden von Pentaprise vergeben
- number - Ganzzahl, Entwicklernummer. Diese Nummer wird auf Anfrage von Pentaprise vergeben.





## 2.2.2 Masken und Screens

### 2.2.2.1 Maskentypen <masktype>

#### ➤ OS\_HOMEPAGE

Definition :

OS\_HOMEPAGE <maskenname>;

Beschreibung:

Loginmaske des ESS. Auf dieser Maske dürfen keine Datenbankfelder verwendet werden. Dieser Typ wird genau einmal benötigt, um die Startmaske im ESS zu definieren.

Parameter :

maskenname - Zeichenkette als eindeutiger Name

#### ➤ OS\_MASK

Definition :

OS\_MASK <maskenname>;

Beschreibung:

Normale Maske mit Eingabefeldern. Von diesem Typ kann es beliebig viele Masken im ESS geben. Auf diesem Maskentyp dürfen alle Feldtypen verwendet werden.

Parameter :

maskenname - Zeichenkette als eindeutiger Name

#### ➤ OS\_MENU

Definition :

OS\_MENU <maskenname>;

Beschreibung:

Maskentyp für Menü's. Von diesem Typ kann es beliebig viele Masken im ESS geben. Sie sollten jedoch nur Felder vom Typ OS\_BUTTON enthalten. Hier dürfen keine Datenbankfelder verwendet werden.

Parameter :

maskenname - Zeichenkette als eindeutiger Name

#### ➤ OS\_MENUBAR

Definition:

OS\_MENUBAR <maskenname>;

Beschreibung:

Maskentyp für Menübar's. Hier sind alle Buttons für eine Menubar zusammengefasst. Hier dürfen keine Datenbankfelder verwendet werden.

Parameter :

maskenname - Zeichenkette als eindeutiger Name.

#### ➤ OS\_SPECMASK

Definition:

OS\_SPECMASK <maskenname>;

Beschreibung:

Maske für besondere Datenbankabfragen durch einen 'speccursor'. Diese Maske muss so viele Felder enthalten, wie die Datenbankabfrage als Rückgabewerte vorsieht.



Parameter :

maskenname - Zeichenkette als eindeutiger Name.

### 2.2.2.2 Maskenauszeichnungen <maskst>

#### ➤ OS\_HIDDEN

Definition :

OS\_HIDDEN;

Beschreibung:

Diese Maske ist eine versteckte Maske, die in keinem Menü vorkommt und auch über keine OS\_POSMASK referenziert wird. Wird hauptsächlich für interne Statistiktabelle verwendet. Unterdrückt die Kompilermeldung 'unreferenced mask'.

#### ➤ OS\_NOWARN

Definition :

OS\_NOWARN;

Beschreibung:

Beim Verlassen der Maske über den Button 'Zurück - ohne Daten' oder 'Zurück - mit Daten' wird keine Überprüfung der Daten vorgenommen, ob diese ohne Speichern verändert worden sind.

#### ➤ OS\_SELUID

Definition :

OS\_SELUID;

Beschreibung:

Der angemeldete Benutzer sieht nur den Datensatz, bei dem in einem Feld vom Typ OS\_UID seine User-ID eingetragen ist.

#### ➤ OS\_SYSPERM

Definition :

OS\_SYSPERM;

Beschreibung:

Belegt die Maske mit 'System'-Permissions d.h. nur der Benutzer 'root' darf schreibend auf dieser Maske arbeiten.

#### ➤ OS\_TITLE

Definition :

OS\_TITLE <titelnummer>;

Beschreibung:

Der Text mit der Titelnummer wird in der Kopfzeile angezeigt. Die Titelnummer wird aus der Datei mxlang verwendet.

Parameter :

titelnummer - Ganzzahl



### ➤ OS\_POSMASK

Definition :

OS\_POSMASK <maskenname>;

Beschreibung:

Referenziert eine Positionsmaske. Da eine Positionsmaske nicht direkt über ein Menü ansprechbar ist, der Maskencompiler jedoch zu jeder Maske einen direkten Zugang erwartet, kann über diesen OS\_POSMASK-Eintrag diese Prüfung ausgeschaltet werden.

Parameter :

maskenname - Zeichenkette, als Maskenname der Positionsmaske.

### ➤ OS\_MMENUBAR

Definition :

OS\_MMENUBAR <maskenname>;

Beschreibung:

Name der Menübar, die oberhalb der Datensatz- oder Detailmaske angezeigt wird. Hier werden automatisch die Buttons aktiv bzw. inaktiv gesetzt.

z.B. Bei Nicht-Datenbankmasken wird der Insert-, Update- und Delete-Button inaktiv gesetzt.

Parameter :

maskenname - Zeichenkette als eindeutiger Name.

### ➤ OS\_IMEXMENUBAR

Definition :

OS\_IMEXMENUBAR <maskenname>;

Beschreibung:

Name der Menübar für Import/Export-Masken. Bewirkt die Aktivierung des Buttons 'Import/Export' auf der OS\_MMENUBAR. Gilt nur für Masken mit Datenbankfeldern.

Parameter :

maskenname - Zeichenkette als eindeutiger Name.

### ➤ OS\_ORDERMENUBAR

Definition :

OS\_ORDERMENUBAR <maskenname>;

Beschreibung:

Name der Menübar für Ordnungs-Masken. Bewirkt die Aktivierung des Buttons 'Ordnen' auf der OS\_MMENUBAR. Gilt nur für Masken mit Datenbankfeldern.

Parameter :

maskenname - Zeichenkette als eindeutiger Name

### ➤ OS\_SELECTMENUBAR

Definition :

OS\_SELECTMENUBAR <maskenname>;

Beschreibung:

Name der Menübar für Abgrenzungs-Masken. Bewirkt die Aktivierung des Buttons 'Abgrenzung' auf der OS\_MMENUBAR. Gilt nur für Masken mit Datenbankfeldern.

Parameter :

maskenname - Zeichenkette als eindeutiger Name



### ➤ OS\_SURVOMENUBAR

Definition :

```
OS_SURVOMENUBAR <maskenname>;
```

Beschreibung:

Name der Menübar für Übersichtspositions-Masken. Bewirkt die Aktivierung des Buttons 'Übersicht setzen' auf der OS\_MMENUBAR. Gilt nur für Masken mit Datenbankfeldern.

Parameter :

maskenname - Zeichenkette als eindeutiger Name

### ➤ OS\_SURVEYMENUBAR

Definition :

```
OS_SURVEYMENUBAR <maskenname>;
```

Beschreibung:

Name der Menübar für Übersichts-Masken. Bewirkt die Aktivierung des Buttons 'Übersicht' auf der OS\_MMENUBAR. Gilt nur für Masken mit Datenbankfeldern.

Parameter :

maskenname - Zeichenkette als eindeutiger Name

## 2.2.2.3 Maskenänderungsanweisungen <updatemaskst>

### ➤ OS\_SORTLIST

Definition :

```
OS_SORTLIST {  
    {  
        <feldname>, <sortierung>;  
        ...  
    }  
}
```

Beschreibung:

Definiert die Standard Sortierreihenfolge auf der Maske, d.h die Datensätze werden immer in dieser Reihenfolge angezeigt.

Parameter :

feldname - Zeichenkette  
sortierung - 0 = Aufsteigend  
            1 = Absteigend

### ➤ OS\_INDEXLIST

Definition :

```
OS_INDEXLIST [ OS_UNIQUE ] {  
    <feldname>, <sortierung>;
```



...

}

Beschreibung:

Erzeugt einen Datenbank-Indexeintrag mit den angegebenen Spalten und der Sortierung.

OS\_UNIQUE erzeugt einen UNIQUE-Index.

Parameter :

Feldname	- Zeichenkette
sortierung	- 0 = Aufsteigend 1 = Absteigend

### ➤ OS\_NMGIVELIST

Definition :

```
OS_NMGIVELIST [ OS_REVERSE ] <listpos> {  
    <maskenname> [ OS_NOMPD ] {  
        <srcfield>, <dstfield> [, <fflag>];  
        ...  
    }  
    OS_SEPARATOR <separatorname>;  
}
```

Beschreibung:

Benennt die Masken, die im Maskenpulldown angezeigt werden.

Das <srcfield> wird in das <dstfield> in der Nextmaske übergeben.

Einträge mit dem Flag OS\_NOMPD erscheinen nicht in dem Maskenpulldown.

Mit der <listpos> wird die Position der OS\_NMGIVELIST bei einer OS\_UPDATEMASK Anweisung angegeben. Innerhalb einer OS\_MASK Anweisung ist die <listpos> irrelevant.

Mit OS\_SEPARATOR kann eine optische Trennung von Masken innerhalb der Maskenauswahl vorgenommen werden. Es wird ein Trennstrich erzeugt.

Mit dem Flag OS\_REVERSE wird die Liste umgedreht, d.h. diese Liste kann in der dst-Maske erstellt werden.

siehe auch OS\_GIVEFIELDLIST.

Parameter :

listpos	- {OS_BEFORE OS_AFTER} {OS_FIRST OS_LAST <maskenname> <separatorname>}
maskenname	- Zeichenkette als Maskenname
srcfield	- Zeichenkette als Feldname
dstfield	- Zeichenkette als Feldname
fflag	- OS_TGSEL, OS_NULL, OS_USERSEL oder OS_COPY
separatorname	- Zeichenkette als eindeutiger Name

### ➤ OS\_NMTAKELIST

Definition :

```
OS_NMTAKELIST [ OS_REVERSE ] {  
    <maskenname> {  
        <dstfield>, <srcfield>;  
    }
```



```
...  
    }  
}
```

**Beschreibung:**

Das <dstfield> aus <maskenname> wird in das <srcfield> in der aktuellen Maske übergeben. Der Übernehmen-Button in der Nextmaske wird aktiviert, wenn man in die Nextmaske wechselt. Mit dem Flag OS\_REVERSE wird die Liste umgedreht, d.h. diese Liste kann in der dst-Maske erstellt werden.

**Parameter :**

maskenname           - Zeichenkette als Maskenname  
srcfield               - Zeichenkette als Feldname  
dstfield               - Zeichenkette als Feldname

➤ **OS\_MFIELDLIST**

**Definition :**

```
OS_MFIELDLIST {  
    OS_DELETE OS_SCREEN <screenid>;  
    <addormove> OS_SCREEN <screenid> OS_TITLE <number> {  
        <addormove> <framename> '[' <xpos>, <ypos> '];  
        ...  
    }  
    ...  
}
```

**Beschreibung:**

Erzeugt mit OS\_ADD einen neuen Screen (Reiter) und ordnet die Frames darauf an. Löscht mit OS\_DELETE einen vorhandenen Screen oder verschiebt mit OS\_MOVE einen vorhandenen Screen an eine andere Position. Mit OS\_ADD werden vorhandene Frames innerhalb des Screen positioniert.

**Parameter :**

screenid           - Ganzzahl, Ordnungsnummer des Screens, d.h. es ist der screenid'te Screen auf der Maske.  
adormove           - OS\_ADD oder OS\_MOVE  
number             - Ganzzahl, eindeutige Titelnummer für diesen Screen.  
framename          - Zeichenkette, Name des Frames  
xpos               - Ganzzahl, X-Position der linken oberen Ecke des Frames in Pixel  
ypos               - Ganzzahl, Y-Position der linken oberen Ecke des Frames in Pixel

## 2.2.2.4 Listenpositionen in Masken <listpos>

➤ **OS\_BEFORE**

**Definition:**

```
OS_BEFORE <listtarget>
```

**Beschreibung:**

Die aktuelle OS\_DEFLIST oder OS\_NMGIVELIST wird vor dem <listtarget> eingefügt. Wirkt nur bei UPDATE-Statements.



➤ **OS\_AFTER**

Definition :

OS\_AFTER <listtarget>

Beschreibung:

Die aktuelle OS\_DEFLIST oder OS\_NMGIVELIST wird nach dem <listtarget> eingefügt. Wirkt nur bei UPDATE-Statements.

➤ **OS\_FIRST**

Definition:

OS\_FIRST

Beschreibung:

Die aktuelle OS\_DEFLIST oder OS\_NMGIVELIST wird an der Ersten Stelle eingefügt. Wirkt nur bei UPDATE-Statements.

➤ **OS\_LAST**

Definition:

OS\_LAST

Beschreibung:

Die aktuelle OS\_DEFLIST oder OS\_NMGIVELIST wird an der Letzten Stelle eingefügt. Wirkt nur bei UPDATE-Statements.

➤ **OS\_NOMPD**

Definition :

OS\_NOMPD

Beschreibung:

Der Maskennamen erscheint nicht im Maskenpulldown, obwohl die OS\_NMGIVELIST ausgeführt wird. Häufig im Zusammenhang mit OS\_COMBI-Feldern verwendet.



## 2.2.3 Frames

### 2.2.3.1 Framedefinition

#### ➤ OS\_FRAME

Definition :

```
OS_FRAME <framename> [ <breite>, <höhe> ] {  
    ...  
}
```

Beschreibung:

Erzeugt einen neuen Frame <framename> in der Dimension <width> x <height>.

Parameter :

framename	- Zeichenkette, als eindeutiger Name des Frames
breite	- Ganzzahl, als Breitenangabe des Frames in Pixel
höhe	- Ganzzahl, als Höhenangabe des Frames in pixel

### 2.2.3.2 Frameauszeichnungen <framest>

#### ➤ OS\_BORDERCOLOR

Definition :

```
OS_BORDERCOLOR <farbwert>;
```

Beschreibung:

Beschreibt die Linienfarbe des Framerahmens in RGB-Notation.

Parameter :

farbwert	- Ganzzahl (RGB) in hexadezimalnotation.
----------	--

#### ➤ OS\_BORDERWIDTH

Definition :

```
OS_BORDERWIDTH <liniendicke>;
```

Beschreibung:

Beschreibt die Dicke der Rahmenlinien in Pixel.

Parameter :

liniendicke	- Ganzzahl, als Liniendicke in Pixel
-------------	--------------------------------------

#### ➤ OS\_BORDERMAP

Definition :

```
OS_BORDERMAP <linien>;
```

Beschreibung:

Definiert die Linien des Rahmens.

Parameter :

linien	- 0 = keine Linien
	1 = obere Linie
	2 = linke Linie
	4 = untere Linie
	8 = rechte Linie





### ➤ OS\_BORDERTYPE

Definition :

OS\_BORDERTYPE <rahmentyp>;

Beschreibung:

Erzeugt einen Rahmen um den Frame mit einem Typ.

Parameter :

rahmentyp                    1 = normale Linie  
                                  2 = hervorgehobener Rahmen  
                                  3 = zurückgesetzter Rahmen

### ➤ OS\_TLCOLOR

Definition :

OS\_TLCOLOR <farbwert>;

Beschreibung:

Beschreibt die Textfarbe des Frametitels in RGB-Notation.

Parameter :

farbwert                    - Ganzzahl (RGB) in hexadezimalnotation.

### ➤ OS\_BGCOLOR

Definition :

OS\_BGCOLOR <farbwert>;

Beschreibung:

Beschreibt die Hintergrundfarbe des Frames in RGB-Notation.

Parameter :

farbwert                    - Ganzzahl (RGB) in hexadezimalnotation.

### ➤ OS\_LINEHEIGHT

Definition :

OS\_LINEHEIGHT <titeltexthöhe>;

Beschreibung:

Beschreibt die Schriftgröße des Frametitels.

Parameter :

titeltexthöhe              - Ganzzahl, in Pixel

### ➤ OS\_TITLE

Definition :

OS\_TITLE <titelnummer>;

Beschreibung:

Der entsprechende Text aus mxlang wird für diesen Frame angezeigt.

Parameter :

titelnummer                - Ganzzahl

### ➤ OS\_TITLEPOS

Definition :

OS\_TILEPOS <anzeige>

Beschreibung:

Der Text aus OS\_TITLE angezeigt.



Parameter :

    anzeige               - 0 = am linken Rahmen  
                          1 = am rechten Rahmen  
                          2 = am oberen Rahmen  
                          3 = am unteren Rahmen

### ➤ OS\_TALIGN

Definition :

    OS\_TALIGN <ausrichtung>;

Beschreibung:

    Der Text aus OS\_TITLE wird ausgerichtet angezeigt.

Parameter :

    ausrichtung       - 0 = links  
                          1 = zentriert  
                          2 = rechts

### ➤ OS\_MFIELDLIST

Definition :

```
OS_MFIELDLIST {  
    OS_DELETE <feldname>;  
    <adormove> <feldname> '[' <xpos>, <ypos> '];  
    ...  
}
```

Beschreibung:

Fügt mit OS\_ADD ein neues Feld in den Frame an der X-Position <xpos> und der Y-Position <ypos> ein. Diese Positionsangaben beziehen sich immer nur auf den Frame.

Löscht mit OS\_DELETE ein vorhandenes Feld aus dem Frame.

Mit OS\_MOVE wird ein vorhandenes Feld an eine andere Position verschoben, jedoch nur in Kombination mit OS\_MOVE des Screens.

#### **ACHTUNG !!**

Sind bereits Positionsänderungen auf der Oberfläche (Editmodus) vorgenommen worden, überlagern diese Einstellungen die Vorgabe aus der Maskenbeschreibung.

Parameter :

    adormove       - OS\_ADD oder OS\_MOVE  
    Feldname       - Zeichenkette, Name des Frames  
    xpos           - Ganzzahl, X-Position der linken oberen Ecke des Frames in Pixel  
    ypos           - Ganzzahl, Y-Position der linken oberen Ecke des Frames in Pixel



### 2.2.3.3 Frameänderungsanweisungen <updateframe>

#### ➤ OS\_UPDATEFRAME

Definition :

```
OS_UPDATEFRAME <framename> '[' <breite>, <höhe> ']' {  
    <framest>  
}
```

Beschreibung:

Verändert die Eigenschaften des Frames <framename>. Bei Angabe von <breite> = 0 und <höhe> = 0 werden die Originaldimensionen nicht verändert.

Parameter :

framename	- Zeichenkette, als eindeutiger Name des Frames
breite	- Ganzzahl, als Breitenangabe des Frames in Pixel
höhe	- Ganzzahl, als Höhenangabe des Frames in pixel



## 2.2.4 Felder

### 2.2.4.1 Felddefinition

```
<fieldtype> Feldname <sizes> <accesslist> {  
    <fieldlistst>  
}
```

### 2.2.4.2 Feldeigenschaften <fieldlist>

#### ➤ OS\_ACTION

Definition :

OS\_ACTION <funktionsname>;

Beschreibung:

Führt angegebene Funktion aus.

Parameter :

ffunc\_back

eine Stufe in der Maskenhierarchie zurück

ffunc\_backtake

eine Stufe in der Maskenhierarchie zurück und den Datensatz entsprechend der Takeliste übernehmen

ffunc\_changepw

Ändern des Benutzerpasswortes

ffunc\_checkpw

Überprüfen des Benutzerpasswortes

ffunc\_clear

Maskenreset (außer Felder mit OS\_EXEC)

ffunc\_dbnext

Positionieren auf den nächsten Datensatz innerhalb der Abgrenzung

ffunc\_dbprev

Positionieren auf den vorherigen Datensatz innerhalb der Abgrenzung

ffunc\_delete

Löschen des aktuellen Datensatzes

ffunc\_detail

Öffnen der Detail-Maske des aktuellen Datensatzes.

ffunc\_euroswitch

Falls für den aktuellen Benutzer eine europäische Landeswährung eingestellt ist, kann die Maskenanzeige zwischen dieser und der Anzeige in EURO gewechselt werden.

ffunc\_exec

Ausführen eines Funktionsformulars

ffunc\_export

Exportieren von Datensätze entsprechend der Abgrenzung und dem Sortierkriterium (Namenskonventionen beachten)

ffunc\_first

Positionieren auf den ersten Datensatz innerhalb der Abgrenzung

ffunc\_goimex

Wechsel zur Import-/Export-Maske

ffunc\_goorder



Öffnen der Ordnungsmaske  
ffunc\_goselect

Öffnen der Abgrenzungsmaske  
ffunc\_gosurvo

Öffnen der Maske zum Einstellen der Übersichtsordnungskriterien.  
ffunc\_help

Öffnen der Maskenhilfe  
ffunc\_home

Rücksprung ins Hauptmenu  
ffunc\_import

Importieren von Datensätzen (Namenskonventionen beachten)  
ffunc\_insert

Einfügen eines Datensatzes  
ffunc\_last

Positionieren auf den letzten Datensatz innerhalb der Abgrenzung  
ffunc\_license

Einfügen neuer ESS-Lizenzen  
ffunc\_logout

Abmelden von System  
ffunc\_next

Wechsel in die entsprechende Datensatzmaske  
ffunc\_order

Aktiviert Ordnungskriterium (nur in Ordnungsmaske)  
ffunc\_pagefirst

Erste Seite in der Übersichtsmaske  
ffunc\_pagelast

Letzte Seite in der Übersichtsmaske  
ffunc\_pagenext

Nächste Seite in der Übersichtsmaske  
ffunc\_pageprev

Vorherige Seite in der Übersichtsmaske  
ffunc\_select

Aktiviert Abgrenzungskriterium (nur in Abgrenzungsmaske)  
ffunc\_selectic

Aktiviert Abgrenzungskriterium ohne Berücksichtigung der Groß- bzw. Kleinschreibung (nur in Abgrenzungsmaske)  
ffunc\_setimex

Öffnet die Maske zum setzen der Spaltenreihenfolge beim Import bzw. Export von Daten.  
ffunc\_setsurvpos

Speichert die gewählte Einstellung für das Aussehen einer Übersichtsmaske benutzerbezogen in der Datenbank.  
ffunc\_survey

Wechsel in die Übersichtsmaske  
ffunc\_this

ohne Funktionalität  
ffunc\_update

Ändern des aktuellen Datensatzes



### ➤ OS\_AICMP

Definition:

OS\_AICMP;

Beschreibung:

Verhindert das Vergleichen des Feldinhaltes beim Einfügen des Datensatzes auf Masken, die mittels OS\_AIPD integriert wurden.

### ➤ OS\_CHECKDEFAULT

Definition:

OS\_CHECKDEFAULT;

Beschreibung:

Prüft den Feldinhalt auf den DEFAULT-Value.

### ➤ OS\_COMBI

Definition :

OS\_COMBI <feldname>;

Beschreibung:

Kombiniert dieses Feld mit einem anderen Feld. Folgende Kombinationen sind möglich:

OS\_TEXT mit OS\_ENUMPD

Es kann von diesem Feld aus in die Masken des OS\_ENUMPD gesprungen werden.

OS\_TEXT mit OS\_GIF

Es wird in OS\_GIF das Image angezeigt, dessen Dateiname in OS\_TEXT angegeben ist.

OS\_TEXT mit OS\_THUMBNAIL

Es wird in OS\_THUMBNAIL das Image als ICON angezeigt, dessen Dateiname in OS\_TEXT angegeben ist.

Parameter :

feldname - Zeichenkette als Feldname

### ➤ OS\_CURCOMBI

Definition :

OS\_CURCOMBI <feldname>;

Beschreibung:

Erstellt eine Währungskette. Das Währungskennzeichen wird neben dem Betrag dargestellt. Die Kette beginnt beim Feld OS\_CURFACT.

Parameter :

feldname - Zeichenkette als Feldname

### ➤ OS\_DBINFO

Definition :

OS\_DBINFO <tabelle>, <spalte>;

Beschreibung:

Beschreibt die Datenbanktabelle und die Datenbankspalte, der dieses Feld zugeordnet ist.

Parameter :

tabelle - Zeichenkette als Tabellennamen

spalte - Zeichenkette als Spaltennamen



### ➤ OS\_DEFAULT

Definition :

```
OS_DEFAULT "<wert>"|fillflag;
```

Beschreibung:

Wert der in die Datenbank geschrieben wird, falls das Feld leer ist.

Parameter :

wert	- String
fillflag	- OS_FILL...

### ➤ OS\_EMPTY

Definition :

```
OS_EMPTY;
```

Beschreibung:

Beim Hinzufügen eines Datensatzes wird dieses Feld geleert. Bei gleichzeitiger Angabe eines OS\_DEFAULT wird immer der Defaultwert beim Einfügen des Datensatzes verwendet.

### ➤ OS\_FKEY

Definition:

```
OS_FKEY;
```

Beschreibung:

Dient dazu dem aktuellen Feld eine Funktionstaste zuzuordnen. Es dürfen Nummer von 1 bis 24 vergeben werden. Wird eine Nummer zwischen 1 und 12 vergeben, kann die dem Feld zugeordnete Aktion direkt über die Funktionstaste ausgelöst werden. Bei Nummer von 13 bis 24 wird zusätzlich ein Modifier (CTRL, SHIFT, ..) benötigt. Die Konfiguration, welcher Modifier verwendet wird, erfolgt über die Benutzereinstellungen.

### ➤ OS\_FORCEDEFAULT

Definition :

```
OS_FORCEDEFAULT "<wert>"|fillflag;
```

Beschreibung:

Wert mit dem das Feld zwangsweise immer gefüllt wird.

Parameter :

wert	- String
fillflag	- OS_FILL...

### ➤ OS\_IMAGE

Definition :

```
OS_IMAGE <imagename>;
```

Beschreibung:

Image, das auf einem Feld vom Typ OS\_BUTTON angezeigt wird.

Parameter :

imagename	- Zeichenkette als Imagename
-----------	------------------------------

### ➤ OS\_ISSURVEY

Definition :

```
OS_ISSURVEY (<stelle>, <länge>);
```



Beschreibung:

Das Feld wird auf der Übersichtsmaske an der Stelle <stelle> mit einer Länge von <länge> in Pixel angezeigt.

Parameter :

stelle - Ganzzahl  
länge - Ganzzahl

### ➤ OS\_JOINFIELD

Definition :

OS\_JOINFIELD <feldname>;

Beschreibung:

Das Feld wird auf der Maske read-only angezeigt, liegt aber physikalisch in einer anderen Tabelle als die aktuelle Maske. Der <feldname> muss eine OS\_REFERENCELIST auf die gejointe Tabelle haben.

Parameter :

feldname - Zeichenkette

### ➤ OS\_NOCHECK

Definition :

OS\_NOCHECK;

Beschreibung:

In Kombination mit OS\_NEXTMASK und OS\_GIVEFIELDLIST. Der Inhalt des Feldes wird bei der Eingabe nicht geprüft. Beim Einfügen in die Datenbank werden aber alle Referenzen berücksichtigt.

### ➤ OS\_NOFILL

Definition :

OS\_NOFILL;

Beschreibung:

Das Feld wird nicht automatisch gefüllt. Nur für OS\_DATE und OS\_TIME.

### ➤ OS\_NOPERMS

Definition :

OS\_NOPERMS;

Beschreibung:

Das Feld kann nicht mit USER-Permissions belegt werden. Es gelten immer die Systempermissions.

### ➤ OS\_SELDEFAULT

Definition :

OS\_SELDEFAULT "<wert>;"

Beschreibung:

Legt ein Standard-Selektionskriterium auf dieses Feld mit diesem <wert>, das beim Betreten dieser Maske aktiv ist.

Parameter :





wert - String

### ➤ OS\_SELF

Definition :

OS\_SELF;

Beschreibung:

Feld für die Eingabe eines Selektionskriteriums auf der eigenen Maske, ohne auf die Selektionsmaske wechseln zu müssen.

### ➤ OS\_TITLE

Definition :

OS\_TITLE <titelnummer> OS\_TITLEPOS <titlepos>;

Beschreibung:

Der Titel mit der Titelnummer (mxlang) wird an der OS\_TITLEPOS vom Feld angezeigt.

Parameter :

titelnummer	- Ganzzahl
titlepos	- 0 = links vom Feld
	1 = rechts vom Feld
	2 = überhalb des Feldes
	3 = unterhalb des Feldes

## 2.2.4.3 Datenbankflags <dbtypeflag>

### ➤ OS\_ALPHACOUNT

Erzeugt einen Eintrag in der Tabelle 'STAB'.

In dieser Tabelle wird ein Nummerkreis für dieses Feld verwaltet, der im Falles eines leeren Eintrags für das Feld zum Einsatz kommt. Ausserdem wird UNIQUE Index für diese Datenbankspalte generiert.

Dieses Feld kann als Referenzfeld verwendet werden (z.B. Adressnummer). Der Nummernkreis wird alphanumerisch gezählt.

### ➤ OS\_GROUP

Nur in Verbindung mit OS\_SERIAL und OS\_JOINFIELD für einen Feldtyp OS\_INT. Erzeugt eine fortlaufende Nummer bezogen auf das OS\_JOINFIELD z.B. Positionsnummer für Auftragspositionen.

### ➤ OS\_NUMCOUNT

Siehe OS\_ALPHACOUNT. Zählt jedoch numerisch.

### ➤ OS\_SERIAL

siehe OS\_GROUP.



### ➤ OS\_UNIQUEDEL

Veranlasst, dass beim Löschen eines Datensatzes vorhandene Notizen und Zusatztexte mit gelöscht werden.

### ➤ OS\_VARCHARTEXT

Falls die verwendete Datenbank den Typ VARCHAR unterstützt, wird diese Spalte als VARCHAR angelegt. Nur sinnvoll bei langen Textfeldern grösser 50 Zeichen.

## 2.2.4.4 Feldeigenschaftsänderungen <updatefieldst>

### ➤ OS\_ACCELL

Definition :

OS\_ACCELL "[<modifier>-]<key>";

Beschreibung:

ShortKey für einen OS\_BUTTON, OS\_FORMPDBUTTON oder OS\_MASKPDBUTTON.

Parameter :

modifier - CTRL, ALT, SHIFT  
key - Tastenbezeichnung z.B. F1

### ➤ OS\_ACCESS

Definition :

OS\_ACCESS <berechtigung>

Beschreibung:

Legt die maximalen Berechtigung für dieses Feld fest. Die Berechtigung können nur nach unten korrigiert werden, jedoch nicht nach oben.

Parameter :

berechtigung - OS\_EXEC, OS\_NOACCESS, OS\_READ oder OS\_WRITE

### ➤ OS\_BGCOLOR

Definition :

OS\_BGCOLOR <Farbwert>;

Beschreibung:

Setzt die Hintergrundfarbe eines Feldes.

Parameter :

Farbwert - Ganzzahl (RGB) in Hexadezimalnotation

### ➤ OS\_DEFLIST

Definition :

OS\_DEFLIST OS\_MASKNAME <listpos>{ <maskenname>, ... }

OS\_DEFLIST <listpos> { <titelnummer>, ... }

Beschreibung:

Enthält die Einträge für ein OS\_ENUMPD oder OS\_ENUMRB. Die Einträge können Titelnummern (mxlang) oder mit der Option OS\_MASKNAME Maskennamen sein.

Parameter :



listpos - siehe OS\_NMGIVELIST  
maskenname - Zeichenkette  
titelnummer - Ganzzahl

### ➤ OS\_DENYNMPD

Definition :

```
OS_DENYNMPD;
```

Beschreibung:

Verhindert das Erzeugen eines Text-Pulldowns.

### ➤ OS\_FALIGN

Definition :

```
OS_FALIGN <ausrichtung>;
```

Beschreibung:

Bestimmt die Ausrichtung des Feldinhaltes.

Parameter :

ausrichtung - 0 = links  
1 = zentriert  
2 = rechts

### ➤ OS\_FGCOLOR

Definition :

```
OS_FGCOLOR <Farbwert>;
```

Beschreibung:

Setzt die Vordergrundfarbe eines Feldes.

Parameter :

Farbwert - Ganzzahl (RGB) als Hexadezimalwert.

### ➤ OS\_FORCE

Definition :

```
OS_FORCE;
```

Beschreibung:

Verhindert, dass dieses Feld leer in die Datenbank geschrieben wird. Es wird auf der Datenbank ein NOT NULL Constraint erzeugt.

### ➤ OS\_GIVEFIELDLIST

Definition :

```
OS_GIVEFIELDLIST {  
    <srcfield>, <dstfield> [, <fflag>];  
    ...  
}
```

Beschreibung:

Feld srcfield wird in dstfield in der OS\_NEXTMASK übergeben.

Mit der Option OS\_TGSEL wird der Wert als Abgrenzungskriterium verwendet.

Mit OS\_NULL wird der Inhalt des Ziel-Feldes geleert.

Mit OS\_USERSEL werden nur Datensätze in der Datenbank gefunden, die vom aktuellen Benutzer angelegt oder zuletzt geändert wurden.



Der Feldtyp und die Feldlänge müssen übereinstimmen.

Parameter :

srcfield - Zeichenkette  
dstfield - Zeichenkette  
fflag - OS\_TGSEL, OS\_NULL, OS\_USERSEL oder OS\_COPY

### ➤ OS\_NEXTMASK

Definition :

OS\_NEXTMASK <maskenname>;

Beschreibung:

Erzeugt den 'Pfeil-nach-unten' nach dem Feld.  
Ermöglicht das Wechseln der Maske nach <maskenname> mit der angegebenen OS\_GIVEFIELDLIST und OS\_TAKEFIELDLIST.

Parameter :

maskenname - Zeichenkette

### ➤ OS\_HEIGHT

Definition :

OS\_HEIGHT <höhe>;

Beschreibung:

Legt die Höhe eines Feldes in Pixel fest. z.B. für Images, Textareas, ...

Parameter :

höhe - Ganzzahl

### ➤ OS\_LINEHEIGHT

Definition :

OS\_LINEHEIGHT <texthöhe>;

Beschreibung:

Beschreibt die Schriftgröße des Feldinhaltes.

Parameter :

texthöhe - Ganzzahl, in Pixel

### ➤ OS\_NOFOCUS

Definition :

OS\_NOFOCUS;

Beschreibung:

Das Feld kann nicht durch normale Tastenbedienung (Enter, TAB) angesprungen werden.

### ➤ OS\_NOBORDER

Definition :

OS\_NOBORDER;

Beschreibung:

Das Feld erhält keinen Rahmen.

### ➤ OS\_NOPOSCHECK

Definition :

OS\_NOPOSCHECK;



Beschreibung:

Die Position des Feldes wird beim Kompilieren nicht geprüft.

#### ➤ OS\_ONELINE

Definition :

OS\_ONELINE;

Beschreibung:

Einzeilige Anzeige von Mehrzeiligen Textfeldern.

#### ➤ OS\_REGEXPR

Definition :

OS\_REGEXPR "<regulärer ausdruck>";

Beschreibung:

Hier können REGULAR-EXPRESSIONS angegeben werden, denen die Eingabe entsprechen muss.

Parameter :

regulärer ausdruck - Zeichenkette

#### ➤ OS\_REFERENCelist

Definition :

OS\_REFERENCelist { <tabelle>, <spalte>; ... }

Beschreibung:

Erzeugt die referenzielle Integrität auf der Datenbank.

Beim Einfügen eines Datensatzes wird geprüft, ob der Wert des Feldes in der <tabelle, spalte> bereits vorhanden ist. Ist der Wert nicht vorhanden, ist keine Datenbankoperation möglich.

Parameter :

tabelle - Zeichenkette

spalte - Zeichenkette

#### ➤ OS\_TAKEFIELDLIST

Definition :

```
OS_TAKEFIELDLIST {  
    <dstfield>, <srcfield>;  
    ...  
}
```

Beschreibung:

Das Feld dstfield wird aus der OS\_NEXTMASK in das srcfield übernommen. Der Feldtyp und die Feldlänge müssen übereinstimmen.

Parameter :

maskenname - Zeichenkette

srcfield - Zeichenkette

dstfield - Zeichenkette

#### ➤ OS\_TALIGN

Definition :

OS\_TALIGN <ausrichtung>;

Beschreibung:



Der Text aus OS\_TITLE wird ausgerichtet angezeigt.

Parameter :

ausrichtung                - 0 = links  
                                  1 = zentriert  
                                  2 = rechts

### ➤ OS\_TLBGCOLOR

Definition :

OS\_TLBGCOLOR <Farbwert>;

Beschreibung:

Setzt die Hintergrundfarbe des Feldtitels.

Parameter :

Farbwert    - Ganzzahl (RGB) als Hexadezimalwert.

### ➤ OS\_TLCOLOR

Definition :

OS\_TLCOLOR <Farbwert>;

Beschreibung:

Setzt die Farbe des Feldtitels.

Parameter :

Farbwert    - Ganzzahl (RGB) als Hexadezimalwert.

### ➤ OS\_TITLEPOS

Definition :

OS\_TITLEPOS <titlepos>;

Beschreibung:

Der Titel wird an der OS\_TITLEPOS vom Feld angezeigt.

Parameter :

titlepos            0 = links vom Feld  
                          1 = rechts vom Feld  
                          2 = überhalb des Feldes  
                          3 = unterhalb des Feldes

### ➤ OS\_TITLESTRING

Definition :

OS\_TITLESTRING;

Beschreibung:

Zeigt den Integerwert des Feldes als Sprachtext an der Oberfläche an.

### ➤ OS\_TITLELEN

Definition :

OS\_TITLELEN <länge>;

Beschreibung:

Der Titel des Feldes wird mit der angegebenen Länge angezeigt.

Parameter :

länge                - Ganzzahl



### ➤ OS\_TITLEHEIGHT

Definition :

OS\_TITLEHEIGHT <höhe>;

Beschreibung:

Der Titel des Feldes wird mit der angegebenen Höhe angezeigt.

Parameter :

höhe - Ganzzahl

### ➤ OS\_TITLEFONT

Definition :

OS\_TITLEFONT fontname;

Beschreibung:

Bestimmt den Schriftfont des Feldtitels;

Parameter:

fontname

### ➤ OS\_TITLEFONTTYPE

Definition :

OS\_TITLEFONTTYPE <zahl>;

Beschreibung:

Bestimmt den Fonttyp des Feldtitels;

Parameter:

zahl -Ganzzahl

### ➤ OS\_VISLEN

Definition :

OS\_VISLEN <länge>;

Beschreibung:

Das Feld wird mit der angegebenen Länge angezeigt.

Parameter :

länge - Ganzzahl

## 2.2.4.5 Feldfüller <fillflags>

### ➤ OS\_FILLACTCUR

Definition :

OS\_FILLACTCUR;

Beschreibung:

Füllt ein Feld mit dem Währungskennzeichen der aktuellen Maskensprache.

### ➤ OS\_FILLACTDATE

Definition :

OS\_FILLACTDATE;

Beschreibung:

Füllt ein OS\_DATE-Feld mit dem aktuellen Datum.



➤ **OS\_FILLACTKW**

Definition :

OS\_FILLACTKW;

Beschreibung:

Füllt ein OS\_INT-Feld mit der aktuellen Kalenderwoche.

➤ **OS\_FILLACTTIME**

Definition :

OS\_FILLACTTIME;

Beschreibung:

Füllt ein OS\_TIME-Feld mit der aktuellen Uhrzeit.

➤ **OS\_FILLLANG**

Definition :

OS\_FILLLANG;

Beschreibung:

Füllt ein OS\_ENUMPD-Feld mit den verfügbaren Sprachen. (mxlangrc)

➤ **OS\_FILLMASKNUM**

Definition :

OS\_FILLMASKNUM;

Beschreibung:

Füllt ein OS\_INT-Feld mit der Titelnummer der aktuellen Maske.

➤ **OS\_FILLTABNAME**

Definition :

OS\_FILLTABNAME;

Beschreibung:

Füllt ein OS\_TEXT-Feld mit dem Namen der aktuellen Maske.

➤ **OS\_FILLUID**

Definition :

OS\_FILLUID;

Beschreibung:

Füllt ein OS\_INT-Feld mit der Login-uid.

➤ **OS\_FILLTELNO**

Definition :

OS\_FILLTELNO;

Beschreibung:

Füllt ein Feld mit einer Telefonnummer. Die Telefonnummer wird aus Informationen in der aktuellen Maske und den Mandanteneinstellung zusammengesetzt. Es gilt folgende Regel:

<Amtsholung>	Feld "AMT" (Mandant)
<Inlandsvorwahl>	Feld "ILK" (Mandant)
<Auslandsvorwahl>	Feld "ALK" (Mandant)
<Landesvorwahl>	Feldtyp "OS_COUNTRYSIGN"





<Vorwahl ohne führende null>  
<Telefonnummer>

Feldtyp "OS\_AREACODE"  
Feldtyp "OS\_FAXNUMBER"

#### 2.2.4.6 Selektionen <tgflag>

##### ➤ OS\_NULL

Definition:

Beschreibung:

##### ➤ OS\_TGSEL

Definition :

OS\_TGSEL

Beschreibung:

Der Wert, der in die Nextmaske übergeben wird, dient als nicht veränderbares Abgrenzungskriterium für diese.

##### ➤ OS\_USERSEL

Definition :

OS\_USERSEL

Beschreibung:

Der Wert, der in die Nextmaske übergeben wird, dient als veränderbares Abgrenzungskriterium für diese.

#### 2.2.4.7 Feldtypen <fieldtype>

##### ➤ OS\_AIPD

Feld Typ für Zusatzmasken. Pulldown wird automatisch gefüllt.

##### ➤ OS\_AREACODE

Feldtyp für die Telefon-Vorwahl. Hauptsächlich für Faxversand. (z.B. für Nürnberg '911'). Siehe auch OS\_TELNO.

##### ➤ OS\_BARPDP

Pulldown zur Anzeige von verfügbaren Barcode eps-Files. Benötigt das Verzeichnis 'barcode' im Homeverzeichnis des ESS-Benutzers.

##### ➤ OS\_BOOL

Boolsches Feld das nur 0 oder 1 enthalten kann.

##### ➤ OS\_BUTTON

Button in einer OS\_MENUBAR oder OS\_MASK. Kann einen Text oder ein Image haben.



➤ **OS\_COUNTRYSIGN**

Landeskennzeichen. (z.B. für Deutschland 'D')

➤ **OS\_CURFACT**

Währungsfaktorfeld. Der Wert eines OS\_CURRENCY-Feldes wird in Abhängigkeit des Wertes der OS\_CURFACT-Feldes dargestellt d.h. ändert sich der Wert des OS\_CURFACT-Feldes ändert sich auch der Wert des OS\_CURRENCY-Feldes. Siehe auch OS\_CURCOMBI.

➤ **OS\_CURRENCY**

Währungsfeld. Zur Darstellung eines Währungsbetrages auf der Oberfläche wird der Datenbankwert des Feldes mit dem Wert des OS\_CURFACT-Feldes multipliziert d.h. in der Datenbank wird nur eine Währung gespeichert. Siehe auch OS\_CURCOMBI.

➤ **OS\_DATE**

Datumfeld. Die Anzeige eines Datumswertes an der Oberfläche geschieht immer in dem für die aktuelle Maskensprache hinterlegtem Format. Auf der Datenbank wird immer das Format 'YYYY-MM-DD' gespeichert. Intern wird dieses Format in Minuten seit 01.01.1800 umgerechnet. Formulare arbeiten immer mit dem Internen Datumsformat als Integer.

➤ **OS\_DIAGPD**

Pulldown zur Anzeige von verfügbaren Diagramm eps-Files. Benötigt das Verzeichnis 'diag' im Homeverzeichnis des ESS-Benutzers.

➤ **OS\_DOUBLE**

Fließkommazahl. Es wird der Fließkommazahltrenner aus den Ländern und die Anzeigegenauigkeit aus der Mandanteneinstellung verwendet.

➤ **OS\_ENUMPD**

Pulldown. Kann Maskennamen oder fixe Werte (mxlang) enthalten.

➤ **OS\_ENUMRB**

Radiobutton (s). Kann nur fixe Werte (mxlang) enthalten.

➤ **OS\_FAXNUMBER**

Feldtyp für die Faxrufnummer. Hauptsächlich für Faxversand. Siehe auch OS\_TELNO.

➤ **OS\_GIF**

Anzeige eines Bildes im Format gif oder jpg. Benötigt einen OS\_COMBI mit einem OS\_TEXT-Feld, in dem der anzuzeigende Dateiname eingetragen ist.

➤ **OS\_HEX**

Ganzzahl in hexadezimal Notation.



➤ **OS\_INT**

Ganzzahl.

➤ **OS\_LABEL**

Der Titeltext des Feldes wird als Label dargestellt.

➤ **OS\_MAIL**

Der Feldtitel wird als ANKER dargestellt. Beim Klick auf das Label des Feldes wird das Mailtool des Browsers aufgerufen (mailto:)

➤ **OS\_MASKPDBUTTON**

Maskenpulldown. Es werden die Maskennamen der Masken angezeigt, zu denen man von dieser Masken aus gelangt.

➤ **OS\_FORMPDBUTTON**

Formularpulldown. Es werden die Formulare angezeigt, die in dem Formscript zu dieser Maske in der Sektion FORMPULLDOWN eingetragen sind.

➤ **OS\_SBOOL**

Selektions-Typ eines Booleschen Wertes. Dieser Feldtyp kann drei Werte verarbeiten:

0 : alle Datensätze in diesem Feld den Wert '0' enthalten

1 : alle Datensätze in diesem Feld den Wert '1' enthalten

% : alle Datensätze

➤ **OS\_SDATE**

Selektions-Typ eines Datums.

➤ **OS\_SHORT**

Kurze Ganzzahl (16 Bit).

➤ **OS\_STATUS**

Abhängig vom Inhalt des Feldes wird der Datensatz auf einer Übersicht in einer benutzerdefinierten Farbe dargestellt.

➤ **OS\_STEXT**

Selektions-Typ eines Textfeldes

➤ **OS\_SUBMASK**

Zeigt eine Übersicht aus der OS\_NEXTMASK an. Die OS\_GIVEFIELDLIST gilt nur für die Anzeige der Datensätze und nicht für den Datenaustausch bei einem Maskenwechsel.

➤ **OS\_TELNO**

Enthält eine zusammengesetzte Telefon-/Telefaxnummer aus OS\_COUNTRYSIGN + OS\_AREACODE + OS\_FAXNUMBER. Siehe auch OS\_FILLTELNO.



➤ **OS\_TEXT**

Textfeld.

➤ **OS\_TEXTAREA**

Textareafeld.

➤ **OS\_THUMBNAIL**

Anzeige eines kleinen Images, dessen File-Name in einem Textfeld eingetragen sein muss und mit dem OS\_THUMBNAIL-Feld kombiniert sein muss. Der Thumbnail wird als Anker dargestellt, der beim betätigen einen Helper aufruft.

➤ **OS\_TIME**

Zeitfeld. Die Anzeige der Zeit an der Oberfläche geschieht im dem Format, das in der aktuellen Maskensprache hinterlegt ist. Die kleinste Zeiteinheit ist Minuten.

➤ **OS\_UID**

In diesem Feld wird die UID des angemeldeten Benutzers angezeigt. Darf nur READ-ONLY sein, da das Feld immer automatisch gefüllt wird.

➤ **OS\_WWW**

Der Feldtitel wird als ANKER dargestellt. Beim Klick auf das Label des Feldes wird ein Browser mit der angegebenen URL aufgerufen.

### 2.2.4.8 Feldberechtigungen <accessval>

➤ **OS\_READ**

Definition :

OS\_READ

Beschreibung:

Das Feld wird READ-ONLY d.h. mit grauem Hintergrund dargestellt.

➤ **OS\_WRITE**

Definition :

OS\_WRITE

Beschreibung:

Alle Eingaben werden als '\*' dargestellt. (z. B. Passwort). In Kombination mit OS\_READ werden die Eingaben in Klartext dargestellt.

➤ **OS\_EXEC**

Definition :

OS\_EXEC

Beschreibung:

Das Feld wird nicht angezeigt.



### ➤ OS\_NOACCESS

Definition :

OS\_NOACCESS

Beschreibung:

wie OS\_EXEC.

## 2.2.4.9 Titelpositionen <titlepos>

### ➤ OS\_PLEFT

Definition:

OS\_PLEFT

Beschreibung:

Das Label zu dem Feld wird links vom Feld dargestellt.

### ➤ OS\_PRIGHT

Definition:

OS\_PRIGHT

Beschreibung:

Das Label zu dem Feld wird rechts vom Feld dargestellt.

### ➤ OS\_PABOVE

Definition:

OS\_PABOVE

Beschreibung:

Das Label zu dem Feld wird überhalb vom Feld dargestellt.

### ➤ OS\_PBELOW

Definition:

OS\_PLEFT

Beschreibung:

Das Label zu dem Feld wird unterhalb vom Feld dargestellt.



## 2.2.5 Formscripten

### 2.2.5.1 Neue Skripteinträge

#### ➤ OS\_CREATEFORMSCRIPT

Definition:

```
OS_CREATEFORMSCRIPT ident <nummer_neu> OS_REPLACE <nummer_alt>;  
OS_CREATEFORMSCRIPT ident <nummer_neu> OS_BEFORE|OS_AFTER <nummer_alt>;  
OS_CREATEFORMSCRIPT ident OS_SEPARATOR separator_ident  
OS_BEFORE|OS_AFTER <nummer_alt>;
```

Beschreibung:

Erzeugt oder ändert einen Formscripteintrag für die Maske ident (die hier als Formscriptdatei kleingeschrieben wird).

Mit OS\_SEPARATOR wird ein Trennstrich erzeugt.

Parameter :

ident	- Name der Formscriptdatei
separator_ident	- Name des Separators
nummer_neu	- Titelnummer des neuen Formscripteintrags
nummer_alt	- Titelnummer eines bestehenden Formscripteintrags, OS_FIRST oder OS_LAST

### 2.2.5.2 Skripteinträge ändern

#### ➤ OS\_UPDATEFORMSCRIPT

Definition:

```
OS_UPDATEFORMSCRIPT ident <scripttype> <formtype> ident_neu <drucker>  
OS_REPLACE ident_alt;  
OS_UPDATEFORMSCRIPT ident <scripttype> <formtype> ident_neu <drucker>  
OS_BEFORE|OS_AFTER ident_alt;
```

Beschreibung:

Fügt ein neues Formular in das Formscript ein oder ersetzt ein bestehendes.

Parameter :

ident	- Name der Formscriptdatei
ident_neu	- Name des neuen Formulars ohne Endung
ident_alt	- Name eines bestehenden Formulars innerhalb des Formscripts
scripttype	- OS_UPDATE OS_INSERT OS_DELETE OS_MASK OS_INPUTACTION OS_FIELD <feldname> OS_FORMPULLDOWN
formtype	- OS_PRINT OS_SURVEYPRINT
drucker	- OS_PRINTER <druckername_aus_ess>



### 3 Formulare

#### Druckformulare:

Ein Druckformular dient dazu Daten aus der Datenbank in formatierter Weise darzustellen. Die somit erzeugten Auswertungen oder Dokumente können dann als PostScript-Dokument ausgedruckt werden (siehe Formularskripten).

Der Formulkompilier "srdbc" dient dazu die Syntax des Druckformularquellcodes zu überprüfen und ein schnell abzuarbeitendes Format zu erzeugen. Die Syntax des Druckformularquellcodes lehnt sich an die Sprache 'C' an, hat aber gegenüber 'C' bestimmte Erweiterungen bzw. Einschränkungen.

#### Maskenformulare:

Maskenformulare sind Programme, die auf Masken lesend und schreibend, auf Datenbankinhalte nur lesend zugreifen können und dadurch bestimmte Funktionen ausführen (z.B. Salden addieren, ...). Sie können jedoch keinen Ausdruck erzeugen !!

Maskenformulare lassen sich nur in Kombination mit einer Lizenz übersetzen.

#### Funktionsformulare:

Funktionsformulare sind Programme, die auf Masken und Datenbankinhalte lesend und schreibend zugreifen können und dadurch bestimmte Funktionen ausführen (z.B. Rechnung erstellen, Rechnung buchen, ...). Sie können jedoch keinen Ausdruck erzeugen !!

Funktionsformulare lassen sich nur in Kombination mit einer Lizenz übersetzen.

### 3.1 Beschreibung der Syntax

<program>	:=	<definition> { <definition> }
		;
<definition>	:=	<ftype> ident ' ( <paramlist> )' <compound>
		<ftype> ident ' ( )' <compound>
		<vardefinition>
		;
<ftype>	:=	'int'   'float'   'text'
		;
ident	:=	[a-zA-Z][_a-zA-Z0-9]*
		;
<paramlist>	:=	{ <type> ident { ',' <type> ident } }
		;
<compound>	:=	'{ { <type> ident ';' } { <statement> } }'
		;
<type>	:=	<ftype>
		'mask' ident
		;
<statement>	:=	'while' ' ( <expr> )' '{ <statement> { <statement> } }'
		'for' ' ( <expr> ';' <expr> ';' <expr> )'
		'{ <statement> { <statement> } }'
		<expr> ';' ;
		'if' ' ( <expr> )'
		'{ <statement> { <statement> } }'
		[ 'else' '{ <statement> { <statement> } }' ]



		'return' <expr> ';' ;
		'break' ',' ;
		'continue' ',' ;
		',' ;
		;
<expr>	:=	<binary>
		;
<binary>	:=	<binary> '=' <binary>
		<binary> '  ' <binary>
		<binary> '&&' <binary>
		<binary> '==' <binary>
		<binary> '!=' <binary>
		<binary> '<' <binary>
		<binary> '<=' <binary>
		<binary> '>' <binary>
		<binary> '>=' <binary>
		<binary> '+' <binary>
		<binary> '-' <binary>
		<binary> '*' <binary>
		<binary> '/' <binary>
		<binary> '%' <binary>
		<unary>
		;
<unary>	:=	'++' <unary>
		'--' <unary>
		'-' <primary>
		'!' <primary>
		<primary>
		;
<primary>	:=	[0-9]*.[0-9]+ (Dezimalzahl in Punktnotation)
		[0-9]+ (Ganzzahl)
		Zeichenkette (in der Form ".....")
		' (' <expr> ')'
		'curcell'
		'globalvector' '[' <primary> ']'
		ident '.' ident
		ident '.' ident '.' 'i' 'o' 's' 'm'
		<primary> '[' <expr> ']'
		ident ' (' { <binary> { ',' <binary> } } )'
		;





## 3.2 Beschreibung der Sprachelemente

### 3.2.1 Funktionen

#### 3.2.1.1 Funktionsdefinition

Eine Funktion besteht aus Funktionstyp, Funktionsname, Parameterliste und Funktionsrumpf. Die Definition erfolgt in der Form:

```
Funktionstyp Funktionsname (Parametertyp Parametername,  
                             Parametertyp Parametername, ...)  
{  
    Variablendefinitionen  
    Anweisungen  
}
```

Als Funktionstyp sind int, float und text erlaubt. Als Parametertyp sind int, float, text und mask erlaubt, wobei mask immer in Verbindung mit einem Maskennamen angegeben werden muss.

#### 3.2.1.2 Besondere Funktionsdefinitionen

Jedes Formular muss genau eine Funktionsdefinition 'main' mit dem Funktionstyp 'int' und einer leeren Parameterliste besitzen. Diese Funktion wird als Startfunktion bei der Ausführung des Formulars verwendet. Außerhalb dieser Funktion dürfen nur globale Variablen und andere Funktionen definiert werden. Neben der Funktion 'main' gibt es noch zwei weitere Funktionen, die besonders behandelt werden und speziell definiert werden müssen, falls die jeweilige Funktionalität benötigt wird. Die Funktionen 'header' und 'footer' haben den Funktionstyp int und besitzen ebenfalls eine leere Parameterliste. Die Funktion 'footer' wird vor einem Seitenwechsel, die Funktion 'header' nach erfolgreichem Seitenwechsel ausgeführt. Innerhalb des Funktionsrumpfes der beiden Funktionen dürfen nur Zelleneigenschaften gesetzt werden und Zuweisungen an Zellen erfolgen.

### 3.2.2 Vordefinierte Funktionen

#### ➤ setcurfieldcont

Definition:

```
int setcurfieldcont (maskfield para_1)
```

Beschreibung :

Setzt den Feldinhalt des Feldes para\_1 der aktuellen Maske, von der aus das Formular gestartet wurde. Der Maskenname der aktuellen Maske und der Maskenname der Maskenvariablen para\_1 müssen übereinstimmen, sonst wird kein Feldinhalt gesetzt.

Wenn Feldinhalte im Rahmen eines Maskenformulars gesetzt werden sollen, muss diese Funktion verwendet werden. Die Funktion ist nur im Funktionsformularkompiler verfügbar.

Rückgabewert:

0



Beispiel :

```
...  
arti.BEZ1_ARTI_ARTI.m = "Neue Artikelbezeichnung";  
setcurfieldcont (arti.BEZ1_ARTI_ARTI);  
...
```

### ➤ **setcurfielderror**

Definition:

```
int setcurfielderror (maskfield para_1)
```

Beschreibung:

Setzt das Fehlerflag des Feldes para\_1 der aktuellen Maske. Das Feld wird auf der Maske rot umrahmt, um den Fehler kenntlich zu machen.

Rückgabewert:

```
0
```

Beispiel:

```
if (arti.BEZ1_ARTI_ARTI.m == "") {  
    setcurfielderror (arti.BEZ1_ARTI_ARTI);  
    errno = 1234;  
    ret = -2;  
}
```

### ➤ **setcurfieldaccess**

Definition:

```
int setcurfieldaccess (text field, int mode)
```

Beschreibung:

Setzt die Feldrechte.

Parameter:

```
field    - Feldname ohne Maskenerweiterung  
mode     - 1 = ausführbar,  
           2 = schreibbar,  
           4 = lesbar
```

Rückgabewert:

```
0
```

Beispiel:

```
...  
if (arti.BEZ1_ARTI_ARTI.m == "") {  
    setcurfieldaccess ("BEZ1_ARTI_ARTI", 6);  
}  
...
```

### ➤ **getcurfieldcont**

Definition:

```
int getcurfieldcont (maskfield para_1)
```

Beschreibung:

Kopiert die Feldinhalte (m,s,o siehe Kapitel 4.3.1 Variablentypen) des Feldes para\_1 der aktuellen Maske, von der aus das Formular gestartet wurde in die Variable para\_1.

Der Maskenname der aktuellen Maske und der Maskenname der Maskenvariablen para\_1 müssen übereinstimmen sonst wird kein Feldinhalt kopiert.



Rückgabewert:

0

Beispiel:

```
...
arti.BEZ1_ARTI_ARTI.m = "Neue Artikelbezeichnung";
setcurfieldcont (arti.BEZ1_ARTI_ARTI);
arti.BEZ1_ARTI_ARTI.m = "";
getcurfieldcont (arti.BEZ1_ARTI_ARTI);
/*arti.BEZ1_ARTI_ARTI.m := "Neue Artikelbezeichnung"*/
...
```

### ➤ **setcurmaskcont**

Definition:

```
int setcurmaskcont (mask para_1)
```

Beschreibung:

Setzt alle Feldinhalte der aktuellen Maske, von der aus das Formular gestartet wurde auf die Feldinhalte der Maskenvariablen para\_1. Der Maskenname der aktuellen Maske und der Maskenname der Maskenvariablen para\_1 müssen übereinstimmen sonst wird kein Feldinhalt übergeben. Die Funktion ist nur im Funktionsformularkompiler verfügbar.

Rückgabewert:

0

Beispiel:

```
...
arti.ART_ARTI_ARTI.m = "ART-00001";
arti.BEZ1_ARTI_ARTI.m = "Neue Artikelbezeichnung";
setcurmaskcont (arti);
```

### ➤ **getcurmaskcont**

Definition:

```
int getcurmaskcont (mask para_1)
```

Beschreibung:

Kopiert alle Feldinhalte der aktuellen Maske, von der aus das Formular gestartet wurde in die Felder der Maskenvariablen para\_1. Der Maskenname der aktuellen Maske und der Maskenname der Maskenvariablen para\_1 müssen übereinstimmen sonst wird kein Feldinhalt kopiert.

Rückgabewert:

0

Beispiel:

```
getcurmaskcont (arti);
```

### ➤ **getmasktitle**

Definition:

```
int getmasktitle (text maskname)
```

Beschreibung:

Liefert die Titelnummer einer Maske zurück. Die Titelnummer ist eine eindeutige Identifizierung der Maske und wird benötigt, um einer Maske (Tabelle) datenbanktechnisch Notizen bzw. Zusatztexte zuzuordnen. Die Zuordnung zu einem bestimmten Datensatz erfolgt über die Titelnummer und eine laufende Datensatznummer.



Rückgabewert:

Maskentitelnummer

Beispiel:

```
tn = getasktitle ("ARTI");
```

### ➤ **sepstrcpy**

Definition:

```
text sepstrcpy (text para_1, text sep)
```

Beschreibung:

Zerlegt einen String "para\_1" mit dem Separator "sep". Der 1. Aufruf erfolgt mit dem zu zerlegenden String. Die nächsten Aufrufe mit einem Leerstring.

Rückgabewert:

Zeichenkette des zerlegten Strings para\_1 oder 0 am Ende.

Beispiel:

```
word = sepstrcpy (string, " ");
while ( (ta_merk = sepstrcpy ("", " ")) != "" ) {
    if (strlen (word + " " + ta_merk) > 100) { break; }
    word = word + " " + ta_merk;
}
cellprops (A_LEFT, F_NORM, SLANT, C_FOR, C_BACK, C_LINE, L_WIDTH, 0, PREC);
nextcell (cell_ypos, cell_xpos, print_width * 100, S_NORM, VSPACE);
curcell = word;
```

### ➤ **cellprops**

Definition:

```
int cellprops (int align, text font, int slant, int fgcolor, int bgcolor, int bcolor,
               int linewidth, int prop, int floatprec)
```

Beschreibung:

Setzt Zellenattribute für alle nachfolgenden Zellenzuweisungen. Es gibt in der Datei 'layout.ssh' bereits vorgefertigte Definements für alle Parameter.

Parameter:

align	- 1 = Text in einer Zelle linksbündig 2 = Text in einer Zelle rechtsbündig 4 = Text in einer Zelle zentriert
font	- Fontname z.B. "Times-Roman" Weitere Fonts finden Sie im Verzeichnis 'formular' in der Datei 'psfonts.ssh'.
slant	- Neigungswinkel des Fonts. Wird nicht ausgewertet !!
fgcolor	- Farbe des Textes in Hexadezimaldarstellung z.B. '0xf0f8ff'. Farbdefinitionen finden Sie im Verzeichnis 'formular' in der Datei 'color.ssh'.
bgcolor	- Hintergrundfarbe der Zelle in Hexadezimaldarstellung. siehe 'fgcolor'.
bcolor	- Rahmenfarbe der Zelle in Hexadezimaldarstellung. siehe 'fgcolor'.
linewidth	-Liniendicke in n-Pixel.
prop	- 0 = Standardeinstellung, keine Linien um die Zelle. 1 = obere Linie einer Zelle darstellen. 2 = untere Linie einer Zelle darstellen. 4 = rechte Linie einer Zelle darstellen. 8 = linke Linie einer Zelle darstellen. 15 = Umrahmung der Zelle. 48 = Unterstreichnung des Textes. 64 = Darstellung eines Graphik eps-Files.



192 = Darstellung eines Diagramm. Das erzeugte eps-File wird gelöscht.

384 = Darstellung eines Barcodes. Das erzeugte eps-File wird gelöscht.

floatprec - Anzahl Nachkommastellen bei Float-Zellen

Rückgabewert:

0

### ➤ makecell

Definition:

int makecell (int y-pos, int x-pos, int cellwidth, int cellheight, int vertspace, int horspace)

Beschreibung:

Setzt Zellenattribute für alle nachfolgenden Zellenzuweisungen. Es gibt in der Datei 'layout.ssh' bereits vorgefertigte definements für alle Parameter.

Parameter:

y-pos - y-Position (Zeile) der linken unteren Ecke der Zelle auf dem Dokument.

x-pos - x-Position (Spalte) der linken unteren Ecke der Zelle auf dem Dokument.

cellwidth - Zellenbreite in 1/10 mm.

cellheight - Zellenhöhe in 1/10 mm. Diese Höhe bestimmt die Schriftgröße in dieser Zelle.

vertspace - vertikaler Abstand zwischen Text und Zellenrand

horspace - horizontaler Abstand zwischen Text und Zellenrand

Rückgabewert:

0

Beispiel:

```
...
cellprops (A_LEFT, F_NORM, SLANT,C_FOR,C_BACK,C_LINE, L_WIDTH, 0, PREC);
makecell (200, 200, 1000, 50, 10);
/* oder als neuer funktionsaufruf
 * nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);
 */
curcell = "Zeile_" + itoa (curline) + " Spalte_1";
...
```

### ➤ newpage

Definition:

int newpage ()

Beschreibung:

Generiert einen Seitenvorschub.

Rückgabewert:

0

Beispiel:

```
...
while (curpage < 5) {
    cellprops (A_LEFT, F_NORM, SLANT,C_FOR,C_BACK,C_LINE, L_WIDTH, 0, PREC);
    nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);
    curcell = "Zeile_" + itoa (curline) + " Spalte_1";
    // hier muss noch eine leere Zelle kommen, damit die letzte Zelle vor dem Seitenumbruch
    // noch auf der vorherigen Seite gedruckt wird !!
    nextcell (cell_ypos, cell_xpos, print_width * 1, S_NORM, VSPACE);
    curcell = "";
    newpage ();
}
```



### ➤ **date2langdate**

Definition:

text date2langdate (int date, text lkz, text format)

Beschreibung:

Wandelt einen internen Datumswert (Minuten seit 01.01.1800) 'date' in eine Zeichenkette nach angegebenem Format 'format' bzw. dem in Landeskennzeichen 'lkz' hinterlegtem Druckformat um. Wird 'format' als leere Zeichenkette übergeben, so wird die im Landeskennzeichen 'lkz' hinterlegte Druckformatangabe verwendet. Wird 'lkz' und 'format' als leere Zeichenkette übergeben, so wird das als Drucksprache angegebene Landeskennzeichen für 'lkz' verwendet.

Rückgabewert:

Zeichenkette in Datumsformat

Beispiel:

```
text datum;
```

```
....
```

```
datum = date2langdate (rek.RDAT_REK_REK.m, "", "%d.%m.%Y");
```

```
/* datum im Format "dd.mm.yyyy" */
```

```
....
```

### ➤ **langdatetodate**

Definition:

int langdatetodate (text indate, text lkz, text format)

Beschreibung:

Ermittelt and Hand des Landeskennzeichens 'lkz' das zugehörige Datumsformat und wandelt 'indate' in ein internes Datumsformat (Minuten seit 01.01.1800) um. Ist 'lkz' nicht angegeben wird 'format' zur Analyse von 'indate' verwendet. Ist auch 'format' nicht angegeben wird die aktuelle Druckeinstellung verwendet.

Rückgabewert:

Integerwert des Datums in Minuten seit 01.01.1800.

Beispiel:

```
int datum;
```

```
....
```

```
datum = langdate2date ("01.05.2000", "", "%d.%m.%Y");
```

```
/* datum im Format "dd.mm.yyyy" */
```

```
....
```

### ➤ **flangstring**

Definition:

text flangstring (int num, text lkz)

Beschreibung:

Gibt den sprachabhängigen Text mit der Nummer 'num' aus der Datei <XESS\_HOME>/etc/mxform zurück. Wird 'lkz' als leere Zeichenkette übergeben, so wird die aktuelle Maskensprache als Drucksprache verwendet.

Rückgabewert:

Zeichenkette.

Beispiel:



....

```
ltext = flangstring (12345, "");
```

....

### ➤ **dateadd**

Definition:

```
int dateadd (int date, int month, int year)
```

Beschreibung:

Addiert auf einen internen Datumswert 'date' Anzahl Monate 'month' plus Anzahl Jahre 'year' .

Rückgabewert:

neuer interner Datumswert

Beispiel:

....

```
newdate = dateadd (date, 2, 0);  
/* newdate = aktuelles Datum plus 2 Monate */
```

....

### ➤ **currency2intern**

Definition:

```
float currency2intern (float value, text wkz, float faktor)
```

Beschreibung:

Wandelt einen angegebenen Währungswert 'value' mit einem angegebenen Währungskennzeichen 'wkz' oder einem angegebenen Faktor 'faktor' in die interne Währungsdarstellung um. Wird als Faktor 'faktor' 0.0 angegeben, so wird der im angegebenen Währungskennzeichen 'wkz' hinterlegte Umrechnungsfaktor zum Berechnen des internen Währungswertes verwendet. Ist als Währungskennzeichen 'wkz' eine leere Zeichenkette angegeben, so wird die aktuelle Maskenwährung als 'wkz' verwendet.

Rückgabewert:

interner Währungswert oder 0 (Fehler)

Beispiel:

....

```
curr_internal = currency2intern (2.5, "DEM", 0.0);  
/* curr_internal = interne Währungsdarstellung von 2,50 DM */  
curr_internal = currency2intern (2.5, "USD", 0.0);  
/* curr_internal = interne Währungsdarstellung von 2,50 US-Dollar */
```

....

### ➤ **intern2currency**

Definition:

```
float intern2currency (float internal, text wkz, float faktor)
```

Beschreibung:

Wandelt einen angegebenen Währungswert 'internal' in interner Währungsdarstellung mit einem angegebenen Währungskennzeichen 'wkz' oder einem angegebenen Faktor 'faktor' in die angegebene Währung um. Wird als Faktor 'faktor' 0.0 angegeben, so wird der im angegebenen Währungskennzeichen 'wkz' hinterlegte Umrechnungsfaktor zum berechnen des darzustellenden Währungswertes verwendet. Ist als Währungskennzeichen 'wkz' eine leere Zeichenkette angegeben, so wird die Währung aus der aktuellen Maskensprache als 'lkz' verwendet.



Rückgabewert:

Landeswährungswert oder 0 (Fehler)

Beispiel:

```
....  
curr_country = intern2currency (2.5, "DEM", 0.0);  
/* curr_country = Landeswährungsdarstellung in DM von intern 2,50 */  
curr_country = intern2currency (2.5, "USD", 0.0);  
/* curr_country = Landeswährungsdarstellung in US-Dollar von intern 2,50 */  
....
```

### ➤ **printcurrency**

Definition:

text printcurrency (float value, text lkz, int print)

Beschreibung:

Wandelt einen angegebenen Währungswert 'value' in mit einem angegebenen Währungskennzeichen 'wkz' in eine Zeichenkette um. Ist als Währungskennzeichen 'wkz' eine leere Zeichenkette angegeben, so wird die aktuelle Maskenwährung als 'wkz' verwendet. Mit der Option 'print' kann die erzeugte Ausgabe gesteuert werden.

Parameter:

value	Währungsbetrag
lkz	Landeskennzeichen
print	0 - Ausgabe der Landeswährung mit landesspezifischer Anzahl von Nachkommastellen und Appendix des Druckwährungszeichens 1 - Ausgabe der Landeswährung mit landesspezifischer Anzahl von Nachkommastellen ohne Appendix des Druckwährungszeichens 2 - nur Ausgabe des Druckwährungszeichens 3 - nur Ausgabe der ISO-Währungsbezeichnung

Rückgabewert:

Zeichenkette der Landeswährungsdarstellung oder leere Zeichenkette (Fehler)

Beispiel:

```
....  
curcell = printcurrency (2.0, "DEM", 0);  
/* curcell = "2,00 DM" */  
curcell = printcurrency (2.0, "USD", 1);  
/* curcell = "2.00" */  
curcell = curcell + printcurrency (0.0, "USD", 3);  
/* curcell = "USD" */  
....
```

### ➤ **uxtime**

Definition:

int uxtime ()

Beschreibung:

Gibt die aktuelle UNIX-Zeit zurück (Sekunden seit 01.01.1970).

Rückgabewert:

aktuelle UNIX-Zeit

Beispiel:

```
....  
timenow = uxtime ();  
....
```





➤ **uxtimes**

Definition:

```
int uxtimes ()
```

Beschreibung:

Gibt die Millisekunden seit dem letzten Systemstart zurück.

Rückgabewert:

Millisekunden seit dem letzten Systemstart

Beispiel:

```
....  
millis_since_boot = uxtimes ();  
....
```

➤ **itoa**

Definition:

```
text itoa (int para_1)
```

Beschreibung:

Wandelt einen Ganzzahlwert para\_1 in eine Zeichenkette um.

Rückgabewert:

Zeichenkette mit numerischem Wert para\_1;

Beispiel:

```
....  
curcell = "Positionsnummer: " + itoa (rep.POS_REP_REP.m);  
....
```

➤ **atoi**

Definition:

```
int atoi (text para_1)
```

Beschreibung:

Wandelt eine numerische Zeichenkette in einen Ganzzahlwert um.

Rückgabewert:

Ganzzahlwert.

Beispiel :

```
....  
textval = "12";  
intvalue = atoi (textval);  
....
```

➤ **ftoa**

Definition:

```
text ftoa (float para_1, int precision, int flag)
```

Beschreibung:

Wandelt einen Floatwert para\_1 in eine Zeichenkette um. 'precision' ist Anzahl der Nachkommastellen. 'flag' 0 nur mit Dezimaltrenner, 1 mit Spracheinstellung.

Rückgabewert:

Zeichenkette mit numerischem Wert para\_1;

Beispiel :

```
....  
curcell = "Lagerbestand: " + ftoa (ap.RBST_AP_AP.m, 2, 1);
```



....

### ➤ **atof**

Definition:

float atof (text para\_1)

Beschreibung:

Wandelt eine numerische Zeichenkette in einen Floatwert um.

Rückgabewert:

Floatwert.

Beispiel :

```
....  
textval = "1.2";  
floatval = atof (textval);  
....
```

### ➤ **round**

Definition:

float round (float value, int prec, int direc)

Beschreibung:

Rundet einen Floatwert 'value' auf 'prec' Nachkommastellen in Richtung 'direc'.

Parameter :

value = zu rundende Fließkommazahl  
prec = es wird auf 'prec' Nachkommastellen gerundet  
direc - 1 = automatisch auf- bzw. abrunden  
2 = aufrunden  
3 = abrunden

Rückgabewert:

gerundete Fließkommazahl.

Beispiel :

```
....  
curcell = "Bestand : " + ftoa (round (ap.RBST_AP_AP.m, 2, 1), 2, 0);  
....
```

### ➤ **mround**

Definition:

float mround (float value, int prec)

Beschreibung:

Rundet eine Fließkommazahl 'value' auf 'prec' Nachkommastellen unter Berücksichtigung von Rundungsfehlern vorheriger mround-Aufrufe. Zurücksetzen erfolgt durch 'value' = 0 und 'prec' = 0. Initialisierung mit einem Rundungsfehler erfolgt durch 'value' = Rundungsfehler und 'prec' = -1.

Parameter:

value = zu rundende Fließkommazahl  
prec = es wird auf 'prec' Nachkommastellen gerundet

Rückgabewert:

gerundete Fließkommazahl;  
aktueller Rundungsfehler beim Zurücksetzen  
0 beim Initialisieren



Beispiel :

```
....  
mround (0, 0);  
f_value = mround (3.147, 2);  
/* f_value = 3.15 */  
f_value = mround (3.146, 2);  
/* f_value = 3.14 */  
....
```

### ➤ lineprops

Definition:

```
int lineprops (int pagebreak)
```

Beschreibung:

Schaltet das Zeilenattribut 'pagebreak' für nachfolgende Zellenzuweisungen ein bzw. aus. 'pagebreak' erlaubt bzw. verbietet einen Seitenumbruch nach dem Drucken einer Zeile.

Parameter:

```
pagebreak -    0 = pagebreak ausschalten  
               1 = pagebreak einschalten
```

Rückgabewert:

```
0
```

Beispiel :

```
....  
nextline (S_NORM);  
lineprops (0);  
cellprops (A_LEFT, F_NORM, SLANT,C_FOR,C_BACK,C_LINE, L_WIDTH, 0, PREC);  
nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);  
curcell = "Text"  
nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);  
curcell = "darf"  
nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);  
curcell = "nicht"  
nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);  
curcell = "umgebrochen"  
nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);  
curcell = "werden"  
nextline (S_NORM);  
lineprops (1);  
nextcell (cell_ypos, cell_xpos, print_width * 10, S_NORM, VSPACE);  
....
```

### ➤ pagesize

Definition:

```
int pagesize (text format)
```

Beschreibung:

Setzt die Papiergröße für den aktuellen Ausdruck. Folgende Papierformate werden erkannt:

"a0"	"a1"	"a2"	"a3"	"a4"	"a5"	"a6"	"a7"
"a8"	"a9"	"a10"	"archA"	"archB"	"archC"	"archD"	"archE"
"b0"	"b1"	"b2"	"b3"	"b4"	"b5"	"flsa"	"flse"



"halfletter"      "ledger"      "legal"      "letter"      "note"

Parameter:

format - s.o.

Rückgabewert:

0

Beispiel :

```
....
pagesize ("letter");
....
```

### ➤ orientation

Definition:

int orientation (int orient)

Beschreibung:

Dreht die Ausgabe auf dem Papier gegen den Uhrzeigersinn.

0 - 0 Grad  
1 - 90 Grad  
2 - 180 Grad  
3 - 270 Grad

Parameter:

orient - 0,1,2,3

Rückgabewert:

0

Beispiel :

```
....
orientation (2); /* erzeugt 'landscape'-Format */
....
```

### ➤ openrw

Definition:

int openrw (text filename, int mode)

Beschreibung:

Öffnet die Datei 'filename' im Modus 'mode'. Es können maximal 20 Dateien gleichzeitig geöffnet sein.

Parameter :

filename            - Pfadname  
mode                - 1 = Datei für Leseoperationen öffnen  
                      2 = Datei neu anlegen und für Schreiboperationen öffnen  
                      3 = Datei für Schreib- und Leseoperationen öffnen

Rückgabewert:

id - Dateibeschreibungsnummer  
-1 - Fehler

Beispiel:

```
....
if ( (file = openrw ("/tmp/myfile", 2)) < 0) {
    ret = -1;
} else {
    if (dbopencursor (arti)) {
        ret = -1;
    }
}
```



```
    } else {
        while (! (i = dbfetchcursor (arti))) {
            line = "";
            line = line + arti.ART_ARTI_ARTI.m + "|";
            line = line + arti.BEZ1_ARTI_ARTI.m + "|";
            writeline (file, line);
        }
        dbclosecursor (arti);
        closerw (file);
    }
}
....
```

### ➤ **closerw**

Definition:

```
int closerw (int file)
```

Beschreibung:

Schließt die Datei mit der Dateibeschreibungsnummer file.

Parameter:

file - Dateibeschreibungsnummer

Rückgabewert:

0 - Datei wurde geschlossen

-1 - Fehler

Beispiel :

```
....
if ( (file = openrw ("/tmp/myfile", 2)) < 0) {
    return -1;
}
....
closerw (file);
....
```

### ➤ **fdprocopen**

Definition:

```
int fdprocopen (text command, text mode)
```

Beschreibung:

Erzeugt eine UNIX-pipe zu dem Kommando 'command' im Modus 'mode'. Abhängig von 'mode' kann diese anschließend mit 'readline' bzw. 'writeline' bearbeitet werden.

Parameter :

filename - Pfadname

mode - "r" = 'pipe' für Leseoperationen öffnen

"w" = 'pipe' für Schreiboperationen öffnen

Rückgabewert:

id - Dateibeschreibungsnummer

-1 - Fehler

Beispiel :

```
....
if ( (file = fdprocopen ("/bin/lis /etc", "r")) < 0) {
    return -1;
}
....
```



```
while (line = readline (file)) {  
    ...  
}  
fdprocclose (file);  
....
```

### ➤ **fdprocclose**

Definition:

```
int fdprocclose (int fd)
```

Beschreibung:

Schließen einer mit 'fdprocopen erzeugten Unix-pipe.

Parameter :

fd - Dateibeschreibungsnummer

Rückgabewert:

-1 - Fehler

sonst - 'exit'-status des ausgeführten UNIX-Befehls.

### ➤ **readline**

Definition:

```
text readline (int file)
```

Beschreibung:

Liest die nächste Zeile aus der Datei mit der Dateibeschreibungsnummer file. Die eingelesene Zeile darf die Zeilenlänge 1023 nicht überschreiten.

Parameter:

file - Dateibeschreibungsnummer

Rückgabewert:

Zeichenkette mit Zeileninhalt

(leere Zeichenkette bei Fehler oder Dateiende);

Beispiel:

```
...  
if ( (file = openrw ("/tmp/myfile", 1)) < 0) {  
    ret = -1;  
} else {  
    while (1) {  
        artikelNr = readline (file);  
        if (!strlen (artikelNr)) {  
            break;  
        }  
        artikelbez = readline (file);  
        if (!strlen (artikelbez)) {  
            break;  
        }  
        arti.ART_ARTI_ARTI.s = artikelNr;  
        if (dbopencursor (arti)) {  
            break;  
        } else {  
            i = dbfetchcursor (arti);  
            if (i) {  
                break;  
            }  
        }  
    }  
}
```



```
    }  
  }  
  dbcclosecursor (arti);  
  arti.BEZ1_ARTI_ARTI.m = artikelbez;  
  ret = dbupdate (arti);  
  if (ret) { break; }  
}  
closerw (file);  
}  
...
```

### ➤ **writeline**

Definition:

int writeline (int file, text line)

Beschreibung:

Schreibt eine Zeile in die Datei mit der Dateibeschreibungsnummer file.

Parameter :

file - Dateibeschreibungsnummer

line - Zeichenkette

Rückgabewert:

0 - Zeile wurde geschrieben

-1 - Fehler

Beispiel:

```
...  
if ( (file = openrw ("/tmp/myfile", 2)) < 0) {  
  ret = -1;  
} else {  
  if (dbopencursor (arti)) {  
    ret = -1;  
  }  
  while (!(i = dbfetchcursor (arti))) {  
    if (writeline (file, arti.ART_ARTI_ARTI.m) < 0) {  
      ret = -1;  
      break;  
    }  
    if (writeline (file, arti.BEZ1_ARTI_ARTI.m) < 0) {  
      rc = -1;  
      break;  
    }  
  }  
}  
dbcclosecursor (arti);  
closerw (file);  
}  
...
```

### ➤ **printdocument**

Definition:

int printdocument ()

Beschreibung:

Startet explizit den Ausdruck eines Dokuments, gibt alle zugewiesenen Zellen frei, und setzt die



Dokumentenvariablen 'curline' und 'curpage' zurück.

Rückgabewert:

0 - kein Fehler

-1 - Fehler

Beispiel :

```
/* Dokument 1 vorbereiten */
curcell = "Dokument 1";
...
/* Dokument 1 ist vorbereitet */
printdocument ();
/* Dokument 2 vorbereiten */
curcell = "Dokument 2";
...
/* Dokument 2 ist vorbereitet */
printdocument ();
...
```

### ➤ **strlen**

Definition:

int strlen (text string)

Beschreibung:

Gibt die Länge der Zeichenkette zurück.

Parameter:

string - Zeichenkette

Rückgabewert:

Länge der Zeichenkette

Beispiel:

```
....
if (strlen (arti.BEZ1_ARTI_ARTI.m)) {
    ....
}
....
```

### ➤ **tempnam**

Definition:

int tempnam ()

Beschreibung:

Erzeugt einen eindeutigen temporären Dateinamen.

Parameter:

keine

Rückgabewert:

Dateiname

Beispiel:

```
....
newfile = tempnam ();
openrw (newfile, 2);
....
```





### ➤ **strcmp**

Definition:

```
int strcmp (text string_1, text string_2)
```

Beschreibung:

Vergleicht die zwei Zeichenketten string\_1 und string\_2 lexikographisch.

Parameter:

string\_1 - Zeichenkette

string\_2 - Zeichenkette

Rückgabewert:

Ganzzahl

< 0 - string\_1 lexikographisch kleiner als string\_2

= 0 - string\_1 ist gleich string\_2

> 0 - string\_1 lexikographisch größer als string\_2

Beispiel :

```
....
if (!(strcmp (arti.ART_ARTI_ARTI.m, "ART-00001"))) {
/* artikelnummer ist "ART-00001" */
    ....
} else {
/* artikelnummer ist nicht "ART-00001" */
    ....
}
```

### ➤ **strncmp**

Definition :

```
int strncmp (text string_1, text string_2, int n)
```

Beschreibung:

Vergleicht die ersten n Zeichen der zwei Zeichenketten string\_1 und string\_2 lexikographisch. (siehe strcmp).

### ➤ **getmxuid**

Definition:

```
int getmxuid ()
```

Beschreibung:

Liefert die Benutzererkennung des angemeldeten Benutzers.

Parameter:

keine.

Rückgabewert:

Benutzererkennung des angemeldeten Benutzers als Integer.

Beispiel :

```
....
uid = getmxuid ();
...

```

### ➤ **getmxlogname**

Definition:

```
int getmxlogname ()
```



Beschreibung:

Liefert den Benutzernamen des angemeldeten Benutzers.

Parameter:

keine.

Rückgabewert:

Benutzernamen des angemeldeten Benutzers.

Beispiel:

```
...  
username = getmxlogname ();  
...
```

### ➤ **getmxhome**

Definition:

```
text getmxhome ()
```

Beschreibung:

Liefert das 'Home'-Verzeichnis des laufenden Daemons 'mxd'.

Parameter :

keine.

Rückgabewert:

Verzeichnisname

Beispiel:

```
...  
esshome = getmxhome ();  
....
```

### ➤ **getflogo**

Definition:

```
text getflogo ()
```

Beschreibung:

Über die Druckeinstellungen oder Benutzereinstellungen können zwei Logo-Dateien konfiguriert werden, die dazu dienen, auf der ersten bzw. allen weiteren Seiten eines Ausdrucks zu erscheinen. Die Funktion liefert den Dateinamen.

Parameter:

keine.

Rückgabewert:

Logodatei

Beispiel :

```
...  
firstlogo = getflogo ();  
....
```

### ➤ **getnlogo**

Definition:

```
text getnlogo ()
```

Beschreibung:

Über die Druckeinstellungen oder Benutzereinstellungen können zwei Logo-Dateien konfiguriert werden, die dazu dienen, auf der ersten bzw. allen weiteren Seiten eines Ausdrucks zu erscheinen. Die Funktion liefert den Dateinamen.



Parameter:

keine.

Rückgabewert:

Logodatei

Beispiel :

```
...
nextlogo = getnlogo ();
....
```

### ➤ **ferrsprintf**

Definition:

```
int ferrsprintf (int num, ...)
```

Beschreibung:

Dient zum Erzeugen von Fehlermeldungen an der Benutzeroberfläche. Mit 'num' wird aus der Datei 'mxerrs' ein Format-String wie in UNIX-printf gesucht. Die weiteren Parameter müssen gemäß dieses Formats korrekt angegeben werden.

Parameter :

num - Fehlernummer aus der Datei 'mxerrs'.

... - wie UNIX-printf

Rückgabewert:

0

Beispiel :

```
...
ferrsprintf (9600, "myfile");
/* Ergebnis: "9600: Datei myfile konnte nicht geöffnet werden." */
....
```

**ACHTUNG: Ein falscher Aufruf kann zum Absturz der Anwendung führen.**

### ➤ **fnotesprintf**

Definition:

```
int fnotesprintf (int num, ...)
```

Beschreibung:

Dient zum Erzeugen von Notizmeldungen an der Benutzeroberfläche. Siehe 'ferrsprintf'.

## 3.2.3 Vordefinierte Datenbankfunktionen

Formulare, die Datenbankänderungen durchführen, werden als Transaktionen behandelt.

In Abhängigkeit von dem Ergebnis des Formulars wird nach Beendigung die Transaktion bestätigt (Returnwert == 0) oder alle Datenbankänderungen werden rückgängig gemacht (Returnwert != 0).

Treten während der Abarbeitung von Formularen Datenbankfehler auf, so werden diese protokolliert und in gekürzter Form in der Statuszeile angezeigt.

### ➤ **dbreadlock**

Definition:



int dbreadlock (mask para\_1)

Beschreibung:

Setzt eine Lesesperre auf alle Tabellen die in der Maske para\_1 enthalten sind. Nach Beendigung des Formulars werden die Sperren automatisch aufgehoben.

Rückgabewert:

0 bei Erfolg

-1 bei Datenbankfehler

Beispiel:

```
...
dbreadlock (arti);
...
```

### ➤ dbopencursor

Definition:

int dbopencursor (mask para\_1)

Beschreibung:

Setzt eine Abgrenzung auf die Maske para\_1 und positioniert den Cursor auf den ersten Datensatz. Der Rückgabewert sagt nichts darüber aus, ob sich Datensätze innerhalb der Abgrenzung befinden.

Rückgabewert:

0 - Erfolg

-1 - Datenbankfehler

Beispiel :

```
arti.ART_ARTI_ARTI.o = 1;
/* 1. Ordnungskriterium ist das Feld ART_ARTI_ARTI */
arti.BEZ1_ARTI_ARTI.o = 2;
/* 2. Ordnungskriterium ist das Feld BEZ1_ARTI_ARTI */

if (dbopencursor (arti)) {
    ret = -1;
} else {
    while (! (i = dbfetchcursor (arti))) {
        /* Zugriff auf Datensatz über artikel.feldname.m */
        ...
        ...
    }
    dbclosecursor (arti);
}
if (i < 0) {
    ret = -1;
}
...
```

### ➤ dbfetchcursor

Definition:

int dbfetchcursor (mask para\_1)

Beschreibung:

Kopiert den Datensatz der aktuellen Cursorposition in die Felder der Maskenvariablen para\_1 (Feldinhalte werden nach Maskenfeldtyp 'm' kopiert), und positioniert auf der Datenbank um



einen Datensatz weiter.

Rückgabewert:

- 0 - Erfolg
- 100 - es befindet sich kein Datensatz an der aktuellen Cursorposition - der Cursor wurde außerhalb der Abgrenzung positioniert oder es existiert kein Datensatz innerhalb der Abgrenzung
- 1 - Datenbankfehler

Beispiel :

siehe dbopencursor

### ➤ **dbclosecursor**

Definition:

int dbclosecursor (mask para\_1)

Beschreibung:

Schließt den Cursor der Maskenvariablen para\_1.

Rückgabewert:

- 0 - Erfolg
- 1 - Datenbankfehler

Beispiel:

siehe dbopencursor

### ➤ **dbdelete**

Definition:

int dbdelete (mask para\_1)

Beschreibung:

Löscht einen Datensatz aus der Haupttabelle von para\_1.  
Es wird automatisch dblock (para\_1) ausgeführt. Die Funktion ist nur im Funktionsformularkompiler verfügbar.

Rückgabewert:

- 0 - Erfolg
- 1 - Datenbankfehler

Beispiel :

```
arti.ART_ARTI_ARTI.s = "ART-00001";  
/* Abgrenzungskriterium ist Artikelnummer, ART-00001 */
```

```
if (dbopencursor (arti)) {  
    return -1;  
} else {  
    if (!(i = dbfetchcursor (arti))) {  
        /* Datensatz aus Tabelle ARTI wird gelöscht */  
        ret = dbdelete (arti);  
    }  
    dbclosecursor (arti);  
    if (rc < 0) {  
        return -1;  
    }  
}
```



### ➤ **dbinsert**

Definition:

```
int dbinsert (mask para_1)
```

Beschreibung:

Fügt einen Datensatz in die Haupttabelle von para\_1 ein. Es wird automatisch dblock (para\_1) ausgeführt. Die Funktion ist nur im Funktionsformularkompiler verfügbar.

Rückgabewert:

```
0 - Erfolg  
-1 - Datenbankfehler
```

Beispiel :

```
/* Sperrkennzeichen ist 0 */  
arti.SPKZ_ARTI_ARTI.m = 0;  
  
/* Datensatz aus der aktuellen Maske wird eingefügt */  
ret = dbinsert (arti);  
setcurmaskcont (arti);  
  
return ret;
```

### ➤ **dbupdate**

Definition:

```
int dbupdate (mask para_1)
```

Beschreibung:

Ändert einen Datensatz in die Haupttabelle von para\_1. Es wird automatisch dblock (para\_1) ausgeführt. Die Funktion ist nur im Funktionscompiler verfügbar.

Rückgabewert:

```
0 - Erfolg  
-1 - Datenbankfehler
```

Beispiel :

```
atrk.ADNR_ATRK_ATRK.m = deb.ADNR_DEB_DEB.m ;  
....  
ret = dbupdate (atrk);
```

## 3.3 Variablen

### 3.3.1 Variablentypen

#### ➤ **Variablentyp int**

Ganzzahlen im Bereich von -2147483647 bis +2147483647 Definition erfolgt in der Form:

```
int varname;
```

Zugriff auf den Inhalt der Variablen erfolgt über den Variablennamen 'varname'.

#### ➤ **Variablentyp float**

Fließkommazahlen im Bereich von -1E76 bis +1E76 Definition erfolgt in der Form:

```
float varname;
```

Zugriff auf den Inhalt der Variablen erfolgt über den Variablennamen 'varname'.



### ➤ Variablentyp text

Zeichenketten beliebiger Länge Definition erfolgt in der Form:

```
text varname;
```

Zugriff auf den Inhalt der Variablen erfolgt über den Variablennamen 'varname'. Zugriff auf das n-te Element (Zeichenkette der Länge 1) einer Zeichenkette erfolgt in der Form:

```
varname[n]
```

Zuweisung an ein Element der Zeichenkette ist nicht gestattet. Mit der Operation '+' können Zeichenketten verknüpft werden.

### ➤ Variablentyp mask

Maske mit den in der Maske befindlichen Feldern. Definition erfolgt in der Form:

```
mask maskename varname;
```

Über den Maskennamen wird definiert welche Felder über die Variable erreichbar sind (Informationen über Maskennamen und in der jeweiligen Maske enthaltene Felder befinden sich in den jeweiligen Masken- bzw. Feldhilfen). Zugriff auf den Inhalt von Maskenfeldern erfolgt über den Variablennamen 'varname' plus 'feldname' plus 'typ' in der Form:

```
varname.feldname.typ
```

Als 'typ' sind 3 unterschiedliche Werte erlaubt:

- m - stellt den Feldinhalt der Maske dar und kann abhängig vom Feldnamen vom typ int, float oder text sein
- s - stellt den Feldinhalt der Maske dar, der bei der Datenbankfunktion 'dbopencursor' als Suchkriterium verwendet wird und ist immer vom Typ 'text'
- o - stellt das Ordnungskriterium dar, das bei der Datenbankfunktion 'dbopencursor' verwendet wird und ist immer vom Typ int (negative Werte bedeuten absteigendes Ordnungskriterium, positive Werte aufsteigendes Ordnungskriterium)

## 3.3.2 Vordefinierte Variablen

### ➤ curcell

Eine besondere Form einer Variablen stellt die aktuelle Zelle 'curcell' dar.

Zellen sind das Instrument mit dem man Informationen auf einer darzustellenden Seite positioniert und mit bestimmten Attributen versehen kann. Zellen müssen nicht eigens definiert werden, sondern sie werden beim Zuweisen automatisch mit dem entsprechenden Typ (int, float oder text) angelegt und ändern bei erneuter Zuweisung automatisch ihren Typ.

Die aktuelle Zelle 'curcell' ist eine globale Variable. Zugriff auf den Inhalt der aktuellen Zelle erfolgt in der Form:

```
curcell = int, float oder text
```

### ➤ globalvector

Eine besondere Form einer Variablen stellt die Variable 'globalvector' dar. Sie ist ein Array von ungetypten Variablen. Erst beim Zuweisen eines Elementes erfolgt die Typisierung des entsprechenden Elements. Zugriff auf den Inhalt des n-ten Globalvectorelements erfolgt in der Form:



globalvector[n, n];

➤ **int curpage**

'curpage' ist eine vordefinierte globale Variable vom Typ int. In ihr wird die aktuelle Seitenzahl mitgezählt.

➤ **int firstpageheader**

Wenn die Variable 'firstpageheader = 1' gesetzt ist, wird die Funktion 'header' auch auf der ersten Seite ausgeführt. 'firstpageheader' ist standardmäßig auf 0 gesetzt.

➤ **int lastpagefooter**

Wenn die Variable 'lastpagefooter = 1' gesetzt ist, wird die Funktion 'footer' auch auf der letzten Seite ausgeführt. 'lastpagefooter' ist standardmäßig auf 0 gesetzt.

➤ **int date**

'date' ist eine vordefinierte globale Variable vom Typ int in der das aktuelle Datum hinterlegt ist.

➤ **int time**

'time' ist eine vordefinierte globale Variable vom Typ int in der die aktuelle Zeit hinterlegt ist.

➤ **int pcurprec**

'pcurprec' ist eine vordefinierte globale Variable vom Typ int in der die Anzahl von Nachkommastellen hinterlegt ist, die in der Spracheinstellung des Benutzers zum drucken von Währungsbeträgen konfiguriert ist.

➤ **int errno**

'errno' ist eine vordefinierte globale Variable vom Typ int, der eine Fehlernummer zugewiesen werden kann. Wenn das Formular ein von 0 verschiedenes Ergebnis liefert wird diese Nummer dazu verwendet eine sprachabhängige Meldung gemäß <IXESS\_HOME>/etc/mxerrs auszugeben. Ein negativer Rückgabewert erzeugt eine Fehlermeldung und ein positiver Rückgabewert erzeugt eine Notizmeldung an der Benutzeroberfläche.





## 4 Perl-Commandos

Perl-Skripte können unter anderem dazu dienen unter dem ESS Abläufe zu automatisieren. Der Import bzw. Export von Datensätzen kann so durchgeführt werden oder jede andere Funktion statt aus der Oberfläche (z.B. über Buttons) durch Aufruf der entsprechenden Funktionen ausgeführt werden.

Die unten abgebildeten Funktionen werden unter `src/programs/perl/` in der Datei `ess.xs` mitgeliefert.

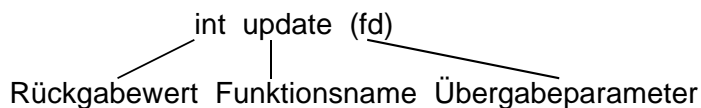
Perl-Beispiel-Programme sind ebenfalls in diesem Verzeichnis zu finden.

### 4.1 Beschreibung der Funktionen

Einem Aufruf der Funktionen muss immer 'ESS:.' vorausgehen. Dies ist eine Klassenreferenz für die Namen der Funktionen.

Die Funktionen sind folgendermaßen aufgebaut:

Bsp:



Datei `ess.xs`:

```
#include "EXTERN.h"
#include "perl.h"
#include "XSUB.h"
```

```
#include <stdio.h>
#include "libif.h"
```

```
MODULE = Ess                PACKAGE = Ess
```

```
PROTOTYPES: DISABLE
```

```
int
update(fd)                  //bewirkt eine Änderung auf der
    int fd                  //Datenbank
    CODE:                   //(entspricht dem Button F2)
        RETVAL = ess_dbupdate(fd);
    OUTPUT:
        RETVAL
```

```
int
delete(fd)                  //Löscht einen (unreferenzierten!)
    int fd                  //Datensatz aus der Datenbank
    CODE:
```



```
        RETVAL = ess_dbdelete(fd);
OUTPUT:
    RETVAL

int
insert(fd)                                //Fügt einen Datensatz in die Daten-
    int fd                                //bank ein
CODE:
    RETVAL = ess_dbinsert(fd);
OUTPUT:
    RETVAL

int
dbnext(fd)                                //wechselt auf den nächsten Daten-
    int fd                                //satz der aktuellen Maske(blättern)
CODE:
    RETVAL = ess_dbnext(fd);
OUTPUT:
    RETVAL

int
backtake(fd, sel)                         //wechselt mit selektierten Daten
    int fd                                //auf den vorherigen Datensatz
    int sel
CODE:
    RETVAL = ess_backtake(fd, sel);
OUTPUT:
    RETVAL

int
godetail(fd, sel)                         //wechselt auf die Maskenansicht des
    int fd                                //ausgewählten Datensatzes
    int sel
CODE:
    RETVAL = ess_godetail(fd, sel);
OUTPUT:
    RETVAL

int
back(fd)                                   //geht ohne Daten zurück
    int fd                                //auf die vorherige Maske
CODE:
    RETVAL = ess_back(fd);
OUTPUT:
    RETVAL
```



```
int
reset(fd)                                     //löscht den Maskeninhalte
  int fd
  CODE:
    RETVAL = ess_reset(fd);
  OUTPUT:
    RETVAL

int
next(fd, fn)                                  //Wechseln über Feldverknüpfung in
  int fd                                       //verknüpften Datensatz.
  char *fn
  CODE:
    RETVAL = ess_next(fd, fn);
  OUTPUT:
    RETVAL

int
formscript(fd, title)                         //ruft das als Parameter übergebene
  int fd                                       // Formscript auf
  int title
  CODE:
    RETVAL = ess_formscript(fd, title);
  OUTPUT:
    RETVAL

int
maskpd(fd, title)                             //entspricht dem Maskenpulldown;
  int fd                                       //wechselt in eine verknüpfte Maske
  int title                                    //als Parameter muss ihre Titel-
  CODE:                                        //nummer angegeben werden
    RETVAL = ess_maskpd(fd, title);
  OUTPUT:
    RETVAL

int
maskpdbyname(fd, mname)                       //siehe maskpd
  int fd                                       //als Parameter kann der Name der
  char *mname                                  //Maske, in die gewechselt werden
  CODE:                                        //soll, angegeben sein
    RETVAL = ess_maskpdbyname(fd, mname);
  OUTPUT:
    RETVAL
```



```
int
setfield(fd, fn, val)           //setzt einen Text in ein Feld
    int fd
    char *fn
    char *val
    CODE:
        RETVAL = ess_setfield(fd, fn, val);
    OUTPUT:
        RETVAL

char *
getfield(fd, fn)               //holt einen Feldtext
    int fd
    char *fn
    CODE:
        RETVAL = ess_getfield(fd, fn);
    OUTPUT:
        RETVAL

int
gopath(fd, path)               //zeigt den Maskenbaum zum aktuellen
    int fd                       //Pfad an
    char *path
    CODE:
        RETVAL = ess_gopath(fd, path);
    OUTPUT:
        RETVAL

int
connect(host, port, user, pass) //bewirkt einen Verbindungsaufbau
    char *host
    hort port
    char *user
    char *pass
    CODE:
        RETVAL = ess_connect(host, port, user, pass);
    OUTPUT:
        RETVAL

void
settracelevel(lev)             //setzt einen Tracelevel für Debug-
    unsigned int lev             //informationen
    CODE:
        ess_settracelevel(lev);

char *
getstack(fd)                   //gibt den aktuellen Pfad (Pointer)
    int fd
    CODE:
        RETVAL = ess_getstack(fd);
    OUTPUT:
        RETVAL

int
select(fd)                     //entspricht dem "go"-Button zur
    int fd                       //Suche nach Datensätzen
    CODE:
        RETVAL = ess_dbselect(fd);
    OUTPUT:
        RETVAL
```



```
int
selectic(fd)                                //entspricht dem case-sensitiven
  int fd                                     //"Go"-Button zur Suche nach
  CODE:                                     //Datensätzen
    RETVAL = ess_dbselectic(fd);
  OUTPUT:
    RETVAL

int
order(fd)                                   //
  int fd
  CODE:
    RETVAL = ess_dborder(fd);
  OUTPUT:
    RETVAL

int
goorder(fd)
  int fd
  CODE:
    RETVAL = ess_dbgoorder(fd);
  OUTPUT:
    RETVAL

int
goselect(fd)                               //entspricht dem Suchen-Button;
  int fd                                     //geht zur Abgrenzungsmaske
  CODE:
    RETVAL = ess_dbgoselect(fd);
  OUTPUT:
    RETVAL

void
disconnect(fd)                             //bewirkt eine Unterbrechnung der
  int fd                                     //Verbindung
  CODE:
    ess_disconnect(fd);
```



```
void
connectinfo(fd)                                //gibt Information über die Ver-
int fd //bindung zum mxd
CODE:
    ess_connectinfo(fd);

char *
error()                                         //gibt Fehlermeldung aus
CODE:
    RETVAL = ess_error();
OUTPUT:
    RETVAL

int
rows(fd)                                       //gibt die selektierte Menge der
int fd                                         //Datensätze aus der Datenbank
CODE:
    RETVAL = ess_rows(fd);
OUTPUT:
    RETVAL

char *
fieldlist(fd, type)                            //listet alle Maskenfelder auf
int fd
int type
CODE:
    RETVAL = ess_fieldlist(fd, type);
OUTPUT:
    RETVAL

char *
apperror(fd)                                   //gibt Fehlermeldung aus Formularen
int fd
CODE:
    RETVAL = ess_apperror(fd);
OUTPUT:
    RETVAL

char *
appnote(fd)                                    //entspricht den Fehlermeldungen
int fd
CODE:
    RETVAL = ess_appnote(fd);
OUTPUT:
    RETVAL
```



```
void
seterrexit( onoff)                                //Globale Variable:
    int onoff                                    //bewirkt ein Programmende im Falle
    CODE:                                        //einer Fehlermeldung
        ess_seterrexit(onoff);

int
maxrows(fd, rows)                                //beschränkt die Datensatzmenge, die
    int fd;                                       //angezeigt werden soll.
    int rows;
    CODE:
        RETVAL = ess_maxrows(fd, rows);
    OUTPUT:
        RETVAL

int
gosurvorder(fd)                                  //geht zur Maske, die eine Übersicht
    int fd;                                       //definiert
    CODE:
        RETVAL = ess_gosurvorder(fd);
    OUTPUT:
        RETVAL

int
survorder(fd)                                    //definiert die Übersicht
    int fd;
    CODE:
        RETVAL = ess_survorder(fd);
    OUTPUT:
        RETVAL

int
survey(fd)                                       //gibt eine Übersicht der aktuellen
    int fd;                                       //Datensätze
    CODE:
        RETVAL = ess_survey(fd);
    OUTPUT:
        RETVAL

int
setsinglemode(fd, mode)                          //je nach Einstellung wird ein Feld-
    int fd;                                       //inhalt sofort an den mxd geschickt
    int mode;                                     //oder er bleibt local
    CODE:
        RETVAL = ess_setsinglemode(fd, mode);
    OUTPUT:
        RETVAL
```



```
char *
getnextfield(fd, flag)                //wechselt ins nächste Maskenfeld
    int fd;
    int flag;
CODE:
    RETVAL = ess_getnextfield(fd, flag);
OUTPUT:
    RETVAL

int
gosubmask(fd, fn, sel)                //wechselt in die Submaske
    int fd;
    char *fn;
    int sel;
CODE:
    RETVAL = ess_gosubmask(fd, fn, sel);
OUTPUT:
    RETVAL
```