

**IntuitionD**

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

	<i>TITLE :</i> IntuitionD		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 24, 2024	

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

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>IntuitionD</b>	<b>1</b>
1.1	IntuitionD . . . . .	1
1.2	TMP:Modula-2/IntuitionD.def . . . . .	3

## Chapter 1

# IntuitionD

### 1.1 IntuitionD

#### Konstanten

altLeft	altRight	amigaKeys
amigaLeft	amigaRight	autoBackPen
autoDrawMode	autoFrontPen	autoITextFont
autoLeftEdge	autoNextText	autoTopEdge
boolGadget	boolMask	buf1024
buf16000	buf2048	buf4096
buf512	buf8000	bufSizeBits
buttonClass	checkImage	checkWidth
close	closeImage	commWidth
correctRGBMask	cursorDown	cursorLeft
cursorRight	cursorUp	custom
customGadget	customImageDepth	customScreen
dModeCount	deadendAlert	defaultMouseQueue
depthImage	dimensionMask	ditheringMask
downImage	driNewLook	driVersion
elite	eoBadformat	eoBigchange
eoClear	eoDelbackward	eoDelforward
eoEnter	eoInsertchar	eoMovecursor
eoNoop	eoReplacechar	eoReset
eoSpecial	eoUndo	euroA0
euroA1	euroA2	euroA3
euroA4	euroA5	euroA6
euroA7	euroA8	eventMax
fanfold	filenameSize	fillrectClass
fine	framefSpecify	frameiClass
frbuttonClass	gRedrawRedraw	gRedrawToggle
gRedrawUpdate	gadgHComp	gadgHNone
gadgHighbits	gadget0002	gadgetClass
gadgetCount	gadgetType	gmDummy
gmGOACTIVE	gmGOINACTIVE	gmHANDLEINPUT
gmHITTEST	gmRENDER	gmrGadgetHit
gmrMeActive	gmrNextActive	gmrNoReuse
gmrPrevActive	gmrReuse	gmrVerify
groupgClass	gzzGadget	highImage
highNone	icClass	icTargetIDCMP
icmCHECKLOOP	icmCLEARLOOP	icmDummy

icmSETLOOP	idsBusy	idsDisabled
idsInactiveDisabled	idsInactiveNormal	←
idsInactiveSelected		
idsIndeterminate	idsNormal	idsSelected
ignoreDimensions	imDraw	imDrawFrame
imErase	imEraseFrame	imFrameBox
imHitFrame	imHitTest	imMove
imageClass	inList	intuitionName
keyCodeB	keyCodeGreater	keyCodeLess
keyCodeM	keyCodeN	keyCodeQ
keyCodeV	keyCodeX	keyCodeZ
knobHmin	knobVmin	lacewb
leftImage	lorientHoriz	lorientNone
lorientVert	lowCheckWidth	lowCommWidth
maxBody	maxPot	maxPubScreenName
menuCancel	menuDown	menuEnabled
menuHot	menuNull	menuUp
menuWaiting	miDrawn	middleDown
middleUp	modelClass	mouseAccel
mxImage	nTractor	noItem
noMenu	noSub	numDriPens
oScanMax	oScanStandard	oScanText
oScanVideo	okAbort	okCancel
okOk	omADDMEMBER	omADDTAIL
omDISPOSE	omDummy	omGET
omNEW	omNOTIFY	omREMMEMBER
omREMOVE	omSET	omUPDATE
opufINTERIM	orderedDithering	osErrNoChipMem
osErrNoChips	osErrNoMem	osErrNoMonitor
osErrPubNotUnique	osErrUnknownMode	otherRefresh
parityMark	parityNone	pica
pointerSize	popPubScreen	propGadget
propgClass	psnPrivate	readBits
recoveryAlert	reqGadget	resCount
rightImage	rootClass	sDownBack
sDragging	sUpFront	scrGadget
screenDrag	sdepthImage	selectDown
selectUp	sgDefaultMaxChars	sgaBeep
sgaEnd	sgaNextActive	sgaPrevActive
sgaRedisplay	sgaReuse	sgaUse
sghClick	sghKey	sgmControl
sgmExitHelp	sgmFixedField	sgmLongint
sgmNoChange	sgmNoFilter	sgmNoWorkb
sgmReplace	shakeXon	shanghai
single	sizeImage	sizing
smartRefresh	stdScreenHeight	stdScreenWidth
stopBits	strGadget	strgClass
sysGadget	sysiClass	sysiaPens
sysisizeHires	sysisizeLowres	sysisizeMedres
topazEighty	topazSixty	upImage
usLegal	usLetter	wDownBack
wDragging	wTractor	wUpFront
wbenchClose	wbenchOpen	writeBits
zoomImage		

Typ-Deklarationen

ActivationFlagSet	ActivationFlags	BaudRate
BoolInfo	BoolInfoPtr	Border
BorderPtr	ClassID	ColorSpec
ColorSpecPtr	DisplayMode	DrawInfo
DrawInfoPtr	DrawPens	EasyStruct
EasyStructPtr	ExtNewScreen	ExtNewScreenPtr
ExtNewWindow	ExtNewWindowPtr	GaTags
Gadget	GadgetFlagSet	GadgetFlags
GadgetInfo	GadgetInfoPtr	GadgetPtr
Gadgets	GpGoInactive	GpGoInactivePtr
GpHitTest	GpHitTestPtr	GpInput
GpInputPtr	GpRender	GpRenderPtr
IBox	IBoxPtr	IClass
IClassPtr	IDCMPFlagSet	IDCMPFlags
IaTags	IcaTags	Image
ImagePtr	ImpDraw	ImpDrawPtr
ImpErase	ImpErasePtr	ImpFrameBox
ImpFrameBoxPtr	ImpHitTest	ImpHitTestPtr
IntuiMessage	IntuiMessagePtr	IntuiText
IntuiTextPtr	IntuitionBase	IntuitionBasePtr
LayoutaTags	Menu	MenuItem
MenuItemFlagSet	MenuItemFlags	MenuItemPtr
MenuPtr	Msg	NewScreen
NewScreenPtr	NewWindow	NewWindowPtr
Object	ObjectPtr	OpAddMember
OpAddMemberPtr	OpAddTail	OpAddTailPtr
OpGet	OpGetPtr	OpMember
OpMemberPtr	OpSet	OpSetPtr
OpUpdate	OpUpdatePtr	PGX
PGXPtr	PgaTags	Preferences
PreferencesPtr	PrintAspect	PrintImage
PrintQuality	PrintShade	PrintSpacing
PrinterFlagSet	PrinterFlags	PrinterPort
PrinterType	PropInfo	PropInfoFlagSet
PropInfoFlags	PropInfoPtr	PubScreenNode
PubScreenNodePtr	Remember	RememberPtr
Requester	RequesterFlagSet	RequesterFlags
RequesterPtr	Res	SGWork
SGWorkPtr	SaTags	Screen
ScreenFlagSet	ScreenFlags	ScreenPtr
SerParShk	SerParShkSet	StrPtr
StringExtend	StringExtendPtr	StringInfo
StringInfoPtr	StringaTags	UObject
UObjectPtr	WaTags	Window
WindowFlagSet	WindowFlags	WindowPtr

## 1.2 TMP:Modula-2/IntuitionD.def

```
DEFINITION MODULE IntuitionD; (*$ Implementation:=FALSE *)
(* 17-Apr-1992/cn *)
```

```
FROM SYSTEM IMPORT ADDRESS, SHORTSET, BITSET, BYTE, CAST, LONGSET;
```

```
FROM ExecD IMPORT
```

```
Interrupt , IOStdReq , Library , List , ListPtr , MemReqSet , Message , MinNode
, MsgPortPtr , SignalSemaphore , Node , TaskPtr ;
```

```
FROM GraphicsD IMPORT
```

```
jam2, BitMap , BitMapPtr , ClipRect , DrawModeSet , GfxBasePtr , LayerInfo
, LayerPtr , Point , RastPort , RastPortPtr , RegionPtr , SimpleSpritePtr
, TextAttr , TextAttrPtr , TextFontPtr , TmpRas , View , ViewModeSet , ViewPort
, ViewPortPtr , ViewPtr ;
```

```
FROM InputEvent IMPORT
```

```
lButton,rButton,upPrefix, InputEvent , InputEventPtr , Qualifiers
, QualifierSet ,mButton;
```

```
FROM KeyMapD IMPORT
```

```
KeyMapPtr ;
```

```
FROM Timer IMPORT
```

```
IOTimer , TimeVal ;
```

```
FROM UtilityD IMPORT
```

```
TagItemPtr ,tagUser, HookPtr , Hook ;
```

```
CONST
```

```
intuitionName="intuition.library";
```

```
TYPE
```

```
StrPtr =ADDRESS;
```

```
(*
```

Die Datentypen von Intuition referenzieren sich oft gegenseitig.  
Aus diesem Grund werden alle Zeiger zuerst deklariert.

```
*)
```

```
BoolInfoPtr =POINTER TO BoolInfo ;
BorderPtr =POINTER TO Border ;
(*36*) ColorSpecPtr =POINTER TO ColorSpec ;
DrawInfoPtr =POINTER TO DrawInfo ;
(*36*) EasyStructPtr =POINTER TO EasyStruct ;
(*36*) ExtNewScreenPtr =POINTER TO ExtNewScreen ;
(*36*) ExtNewWindowPtr =POINTER TO ExtNewWindow ;
(*36*) GadgetInfoPtr =POINTER TO GadgetInfo ;
GadgetPtr =POINTER TO Gadget ;
(*36*) GpGoInactivePtr =POINTER TO GpGoInactive ;
(*36*) GpHitTestPtr =POINTER TO GpHitTest ;
(*36*) GpInputPtr =POINTER TO GpInput ;
(*36*) GpRenderPtr =POINTER TO GpRender ;
(*36*) IBoxPtr =POINTER TO IBox ;
(*36*) IClassPtr =POINTER TO IClass ;
ImagePtr =POINTER TO Image ;
(*36*) ImpDrawPtr =POINTER TO ImpDraw ;
(*36*) ImpErasePtr =POINTER TO ImpErase ;
(*36*) ImpFrameBoxPtr =POINTER TO ImpFrameBox ;
(*36*) ImpHitTestPtr =POINTER TO ImpHitTest ;
IntuiMessagePtr =POINTER TO IntuiMessage ;
IntuiTextPtr =POINTER TO IntuiText ;
IntuitionBasePtr =POINTER TO IntuitionBase ;
MenuItemPtr =POINTER TO MenuItem ;
MenuPtr =POINTER TO Menu ;
```

---

```

NewScreenPtr =POINTER TO NewScreen ;
NewWindowPtr =POINTER TO NewWindow ;
(*36*) OpGetPtr =POINTER TO OpGet ;
(*36*) ObjectPtr =POINTER TO Object ;
(*36*) OpAddTailPtr =POINTER TO OpAddTail ;
(*36*) OpSetPtr =POINTER TO OpSet ;
(*36*) OpMemberPtr =POINTER TO OpMember ;
(*36*) OpUpdatePtr =POINTER TO OpUpdate ;
(*36*) PGXPtr =POINTER TO PGX ;
PreferencesPtr =POINTER TO Preferences ;
PropInfoPtr =POINTER TO PropInfo ;
(*36*) PubScreenNodePtr =POINTER TO PubScreenNode ;
RememberPtr =POINTER TO Remember ;
RequesterPtr =POINTER TO Requester ;
ScreenPtr =POINTER TO Screen ;
(*36*) SGWorkPtr =POINTER TO SGWork ;
(*36*) StringExtendPtr =POINTER TO StringExtend ;
StringInfoPtr =POINTER TO StringInfo ;
(*36*) UObjectPtr =POINTER TO UObject ;
WindowPtr =POINTER TO Window ;

```

CONST

```

(*)
  Bitnummern für Menu.flags:
*)
menuEnabled=0;
miDrawn=8;

```

TYPE

```

Menu =RECORD
  nextMenu: MenuPtr ;
  leftEdge:INTEGER;
  topEdge:INTEGER;
  width:INTEGER;
  height:INTEGER;
  flags:BITSET;
  menuName:ADDRESS;
  firstItem: MenuItemPtr ;
  jazzX:INTEGER;
  jazzY:INTEGER;
  beatX:INTEGER;
  beatY:INTEGER;

```

END;

```

MenuItemFlags =(
  checkIt,itemText,commSeq,menuToggle,itemEnabled,mif5,
  highComp,highBox,checked,mif9,mif10,mif11,isDrawn,
  highItem,menuToggled,mif15
);
MenuItemFlagSet =SET OF MenuItemFlags ;

```

CONST

```

highImage= MenuItemFlagSet {};
highNone= MenuItemFlagSet {highBox,highComp};

```

TYPE

```

MenuItem =RECORD

```

---



```

nextItem: MenuItemPtr ;
leftEdge:INTEGER;
topEdge:INTEGER;
width:INTEGER;
height:INTEGER;
flags: MenuItemFlagSet ;
mutualExclude:LONGSET;
itemFill:ADDRESS;
selectFill:ADDRESS;
command:CHAR;
subItem: MenuItemPtr ;
nextSelect:CARDINAL;
END;

```

```

RequesterFlags =(
pointRel,preDrawn,noisyReq,rf3,
simpleReq,(*36*)useReqImage,noReqBackFill,
rf7,rf8,rf9,rf10,rf11,
reqOffWindow,reqActive,sysRequest,deferRefresh
);
RequesterFlagSet =SET OF RequesterFlags ;

```

```

Requester =RECORD
olderRequest: RequesterPtr ;
leftEdge:INTEGER;
topEdge:INTEGER;
width:INTEGER;
height:INTEGER;
relLeft:INTEGER;
relTop:INTEGER;
reqGadget: GadgetPtr ;
reqBorder: BorderPtr ;
reqText: IntuiTextPtr ;
flags: RequesterFlagSet ;
backFill:SHORTCARD;
reqLayer: LayerPtr ;
reqPad1:ARRAY [0..31] OF BYTE;
imageBMap: BitMapPtr ;
rWindow: WindowPtr ;
(*36*)reqImage: ImagePtr ;
reqPad2:ARRAY [0..31] OF BYTE;
END;

```

```

GadgetFlags =(
gadgHBox,gadgHImage,gadgImage,gRelBottom,gRelRight,
gRelWidth,gRelHeight,selected,gadgDisabled,
(*37*)tabCycle,(*37*)stringExtend,gf11,
(*36*)labelString,(*36*)labelImage,
gf14,gf15
);
GadgetFlagSet =SET OF GadgetFlags ;

```

```

ActivationFlags =(
relVerify,gadgImmediate,endGadget,followMouse,
rightBorder,leftBorder,topBorder,bottomBorder,
toggleSelect,stringCenter,stringRight,longint,
altKeyMap,boolExtend,activeGadget,borderSniff

```

---

```

);
  ActivationFlagSet =SET OF  ActivationFlags ;

CONST
  gadgHighbits=CAST( GadgetFlagSet ,03H);
  gadgHNone= GadgetFlagSet {gadgHBox,gadgHImage};
  gadgHComp= GadgetFlagSet {};

  boolGadget=0001H;
  gadget0002=0002H;
  propGadget=0003H;
  strGadget=0004H;
  (*36*) customGadget=0005H;

  sizing=0010H;
  wDragging=0020H;
  sDragging=0030H;
  wUpFront=0040H;
  sUpFront=0050H;
  wDownBack=0060H;
  sDownBack=0070H;
  close=0080H;

  reqGadget=1000H;
  gzzGadget=2000H;
  scrGadget=4000H;
  sysGadget=8000H;
  gadgetType=CAST (BITSET,0FC00H) ;

TYPE
  Gadget =RECORD
    nextGadget: GadgetPtr ;
    leftEdge:INTEGER;
    topEdge:INTEGER;
    width:INTEGER;
    height:INTEGER;
    flags: GadgetFlagSet ;
    activation: ActivationFlagSet ;
    gadgetType:CARDINAL;
    gadgetRender:ADDRESS;
    selectRender:ADDRESS;
    gadgetText: IntuiTextPtr ;
    CASE :INTEGER OF
      |0: mutualExclude:LONGSET;
      |1: (*36*)hp: HookPtr ;
    END;
    specialInfo:ADDRESS;
    gadgetID:INTEGER;
    userData:ADDRESS;
  END;

CONST
  (*
  Bitnummer für BoolInfo.flags.
  *)
  boolMask=0;

```

---

```

TYPE
  BoolInfo =RECORD
    flags:BITSET;
    mask:ADDRESS;
    reserved:LONGCARD;
  END;

  PropInfoFlags =(
    autoKnob,freeHoriz,freeVert,propBorderless,
    (*36*)propNewLook,pf5,pf6,pf7,
    knobHit,pf9,pf10,pf11,pf12,pf13,pf14,pf15
  );
  PropInfoFlagSet =SET OF  PropInfoFlags ;

```

```

CONST
  knobVmin=4;
  knobHmin=6;
  maxBody=0FFFFH;
  maxPot=0FFFFH;

```

```

TYPE
  PropInfo =RECORD
    flags: PropInfoFlagSet ;
    horizPot:CARDINAL;
    vertPot:CARDINAL;
    horizBody:CARDINAL;
    vertBody:CARDINAL;
    cWidth:CARDINAL;
    cHeight:CARDINAL;
    hPotRes:CARDINAL;
    vPotRes:CARDINAL;
    leftBorder:CARDINAL;
    topBorder:CARDINAL;
  END;

```

```

TYPE
  StringInfo =RECORD
    buffer:ADDRESS;
    undoBuffer:ADDRESS;
    bufferPos:INTEGER;
    maxChars:INTEGER;
    dispPos:INTEGER;
    undoPos:INTEGER;
    numChars:INTEGER;
    dispCount:INTEGER;
    cLeft:INTEGER;
    cTop:INTEGER;
    (*36*)extension: StringExtendPtr ;
    longInt:LONGINT;
    altKeyMap: KeyMapPtr ;
  END;

```

```

CONST
  autoFrontPen=0;
  autoBackPen=1;
  autoDrawMode=jam2;
  autoLeftEdge=6;

```

---

```

autoTopEdge=3;
autoITextFont=NIL;
autoNextText=NIL;

```

```
TYPE
```

```

  IntuiText =RECORD
    frontPen:SHORTCARD;
    backPen:SHORTCARD;
    drawMode: DrawModeSet ;
    leftEdge:INTEGER;
    topEdge:INTEGER;
    iTextFont: TextAttrPtr ;
    iText:ADDRESS;
    nextText: IntuiTextPtr ;
  END;

```

```

  Border =RECORD
    leftEdge:INTEGER;
    topEdge:INTEGER;
    frontPen:SHORTCARD;
    backPen:SHORTCARD;
    drawMode: DrawModeSet ;
    count:SHORTCARD;
    xy:ADDRESS;
    nextBorder: BorderPtr ;
  END;

```

```

  Image =RECORD
    leftEdge:INTEGER;
    topEdge:INTEGER;
    width:INTEGER;
    height:INTEGER;
    depth:INTEGER;
    imageData:ADDRESS;
    planePick:SHORTCARD;
    planeOnOff:SHORTCARD;
    nextImage: ImagePtr ;
  END;

```

```

  IDCMPFlags =(
    sizeVerify,newSize,refreshWindow,mouseButtons,
    mouseMove,gadgetDown,gadgetUp,reqSet,
    menuPick,closeWindow,rawKey,reqVerify,
    reqClear,menuVerify,newPrefs,diskInserted,
    diskRemoved,whenchMessage,activeWindow,inactiveWindow,
    deltaMove,vanillaKey,intuiTicks,(*36*)idcmpUpdate,
    (*36*)menuHelp,(*36*)changeWindow,c26,c27,
    c28,c29,c30,lonlyMessage
  );
  IDCMPFlagSet =SET OF  IDCMPFlags ;

```

```
CONST
```

```

  selectUp=lButton+upPrefix;
  selectDown=lButton;
  menuUp=rButton+upPrefix;
  menuDown=rButton;
  middleDown=mButton;

```

---

```
middleUp=mButton+upPrefix;

checkWidth=19;
commWidth=27;
lowCheckWidth=13;
lowCommWidth=16;

menuNull=0FFFFH;
noMenu=1FH;
noItem=3FH;
noSub=1FH;

keyCodeQ=10H;
keyCodeZ=31H;
keyCodeX=32H;
keyCodeV=34H;
keyCodeB=35H;
keyCodeN=36H;
keyCodeM=37H;
keyCodeLess=38H;
keyCodeGreater=39H;

cursorUp=4CH;
cursorDown=4DH;
cursorRight=4EH;
cursorLeft=4FH;

menuHot=1;
menuCancel=2;
menuWaiting=3;

okOk=menuHot;
okAbort=4;
okCancel=menuCancel;

wbenchOpen=1;
wbenchClose=2;

deadendAlert=80000000H;
recoveryAlert=0;

altLeft= QualifierSet {lAlt};
altRight= QualifierSet {rAlt};
amigaLeft= QualifierSet {lCommand};
amigaRight= QualifierSet {rCommand};
amigaKeys=amigaLeft+amigaRight;

TYPE
  IntuiMessage =RECORD
    execMessage: Message ;
    class: IDCMPFlagSet ;
    code:CARDINAL;
    qualifier: QualifierSet ;
    iAddress:ADDRESS;
    mouseX:INTEGER;
    mouseY:INTEGER;
    seconds:LONGCARD;
```

---

```

micros:LONGCARD;
idcmpWindow: WindowPtr ;
specialLink: IntuiMessagePtr ;
END;

```

```

IBox =RECORD
left:INTEGER;
top:INTEGER;
width:INTEGER;
height:INTEGER;
END;

```

TYPE

```

WindowFlags =(
windowSizing,windowDrag,windowDepth,windowClose,
sizeBRight,sizeBBottom,simpleRefresh,superBitMap,
backDrop,reportMouse,gimmeZeroZero,borderless,
activate,windowActive,inRequest,menuState,
rmbTrap,noCareRefresh,
(*36*)nwExtended,wf19,wf20,wf21,wf22,wf23,
windowRefresh,wbenchWindow>windowTicked,
(*36*)visitor,(*36*)zoomed,(*36*)hasZoom,wf30,wf31
);
WindowFlagSet =SET OF WindowFlags ;

```

```

ScreenFlags =(
wbenchScreen,(*36*)publicScreen,sf2,sf3,
showTitle,beeping,customBitMap,screenBehind,
screenQuiet,screenHires,sf10,sf11,
nsExtended,sf13,autoScroll,sf15
);
ScreenFlagSet =SET OF ScreenFlags ;

```

CONST

```

(*36*)defaultMouseQueue=5;
smartRefresh= WindowFlagSet {};
otherRefresh= WindowFlagSet {simpleRefresh,superBitMap};

```

TYPE

```

Window =RECORD
nextWindow: WindowPtr ;
leftEdge:INTEGER;
topEdge:INTEGER;
width:INTEGER;
height:INTEGER;
mouseY:INTEGER;
mouseX:INTEGER;
minWidth:INTEGER;
minHeight:INTEGER;
maxWidth:INTEGER;
maxHeight:INTEGER;
flags: WindowFlagSet ;
menuStrip: MenuPtr ;
title:ADDRESS;
firstRequest: RequesterPtr ;
dmRequest: RequesterPtr ;
reqCount:INTEGER;

```

```

wScreen: ScreenPtr ;
rPort: RastPortPtr ;
borderLeft: SHORTINT;
borderTop: SHORTINT;
borderRight: SHORTINT;
borderBottom: SHORTINT;
borderRPort: RastPortPtr ;
firstGadget: GadgetPtr ;
parent: WindowPtr ;
descendant: WindowPtr ;
pointer: ADDRESS;
ptrHeight: SHORTINT;
ptrWidth: [0..16];
xOffset: SHORTINT;
yOffset: SHORTINT;
idcmpFlags: IDCMPFlagSet ;
userPort: MsgPortPtr ;
windowPort: MsgPortPtr ;
messageKey: IntuiMessagePtr ;
detailPen: SHORTCARD;
blockPen: SHORTCARD;
checkMark: ImagePtr ;
screenTitle: ADDRESS;
gzzMouseX: INTEGER;
gzzMouseY: INTEGER;
gzzWidth: INTEGER;
gzzHeight: INTEGER;
extData: ADDRESS;
userData: ADDRESS;
wLayer: LayerPtr ;
iFont: TextFontPtr ;
(*36*) moreFlags: LONGSET;
END;

```

```

NewWindow =RECORD
leftEdge: INTEGER;
topEdge: INTEGER;
width: INTEGER;
height: INTEGER;
detailPen: SHORTINT;
blockPen: SHORTINT;
idcmpFlags: IDCMPFlagSet ;
flags: WindowFlagSet ;
firstGadget: GadgetPtr ;
checkMark: ImagePtr ;
title: ADDRESS;
screen: ScreenPtr ;
bitMap: BitMapPtr ;
minWidth: INTEGER;
minHeight: INTEGER;
maxWidth: INTEGER;
maxHeight: INTEGER;
type: ScreenFlagSet ;
END;

```

```

(*36*) ExtNewWindow =RECORD
nw: NewWindow ;

```

```
        extension: TagItemPtr ;
    END;

(*36*) WaTags =(waDummy:=tagUser+99,
    waLeft,
    waTop,
    waWidth,
    waHeight,
    waDetailPen,
    waBlockPen,
    waIDCMP,
    waFlags,
    waGadgets,
    waCheckMark,
    waTitle,
    waScreenTitle,
    waCustomScreen,
    waSuperBitMap,
    waMinWidth,
    waMinHeight,
    waMaxWidth,
    waMaxHeight,
    waInnerWidth,
    waInnerHeight,
    waPubScreenName,
    waPubScreen,
    waPubScreenFallBack,
    waWindowName,
    waColors,
    waZoom,
    waMouseQueue,
    waBackFill,
    waRptQueue,
    waSizeGadget,
    waDragBar,
    waDepthGadget,
    waCloseGadget,
    waBackdrop,
    waReportMouse,
    waNoCareRefresh,
    waBorderless,
    waActivate,
    waRMBTrap,
    waWBenchWindow,
    waSimpleRefresh,
    waSmartRefresh,
    waSizeBRight,
    waSizeBBottom,
    waAutoAdjust,
    waGimmeZeroZero,
    (*37*) waMenuHelp
);

CONST
    driVersion=1;

TYPE
```

---



```

    DrawPens =(
    detailPen,blockPen,textPen,shinePen,shadowPen,fillPen,fillTextPen
    ,backGroundPen,highLightTextPen
    );

CONST
    numDriPens=ORD (MAX ( DrawPens ))+1;

(*
    Bitnummer für DrawInfo.flags
*)
    driNewLook=0;

TYPE
    DrawInfo =RECORD
        version:CARDINAL;
        numPens:CARDINAL;
        pens:POINTER TO ARRAY DrawPens OF CARDINAL;
        font: TextFontPtr ;
        depth:CARDINAL;
        resolution:RECORD
            x,y:CARDINAL;
        END;
        flags:LONGSET;
        reserved:ARRAY[0..6] OF LONGCARD;
    END;

CONST
    stdScreenHeight=-1;
    (*36*)stdScreenWidth=-1;

    customScreen= ScreenFlagSet {wbenchScreen..sf3};

TYPE
    Screen =RECORD
        nextScreen: ScreenPtr ;
        firstWindow: WindowPtr ;
        leftEdge:INTEGER;
        topEdge:INTEGER;
        width:INTEGER;
        height:INTEGER;
        mouseY:INTEGER;
        mouseX:INTEGER;
        flags: ScreenFlagSet ;
        title:ADDRESS;
        defaultTitle:ADDRESS;
        barHeight:SHORTINT;
        barVBorder:SHORTINT;
        barHBorder:SHORTINT;
        menuVBorder:SHORTINT;
        menuHBorder:SHORTINT;
        wBorTop:SHORTINT;
        wBorLeft:SHORTINT;
        wBorRight:SHORTINT;
        wBorBottom:SHORTINT;
        font: TextAttrPtr ;
        viewPort: ViewPort ;

```

---

```
    rastPort: RastPort ;
    bitMap: BitMap ;
    layerInfo: LayerInfo ;
    firstGadget: GadgetPtr ;
    detailPen:SHORTCARD;
    blockPen:SHORTCARD;
    saveColor0:CARDINAL;
    barLayer: LayerPtr ;
    extData:ADDRESS;
    userData:ADDRESS;
END;
```

```
NewScreen =RECORD
    leftEdge:INTEGER;
    topEdge:INTEGER;
    width:INTEGER;
    height:INTEGER;
    depth:INTEGER;
    detailPen:SHORTINT;
    blockPen:SHORTINT;
    viewModes: ViewModeSet ;
    type: ScreenFlagSet ;
    font: TextAttrPtr ;
    defaultTitle:ADDRESS;
    gadgets: GadgetPtr ;
    customBitMap: BitMapPtr ;
END;
```

```
(*36*) ExtNewScreen =RECORD
    ns: NewScreen ;
    extension: TagItemPtr
END;
```

```
(*36*) SaTags =(saDummy:=tagUser+32,
    saLeft,
    saTop,
    saWidth,
    saHeight,
    saDepth,
    saDetailPen,
    saBlockPen,
    saTitle,
    saColors,
    saErrorCode,
    saFont,
    saSysFont,
    saType,
    saBitMap,
    saPubName,
    saPubSig,
    saPubTask,
    saDisplayID,
    saDClip,
    saOverscan,
    saObsoletel,
    saShowTitle,
    saBehind,
```

---

```

    saQuiet,
    saAutoScroll,
    saPens,
    saFullPalette
);

```

CONST

```

(*36*) osErrNoMonitor=1;
(*36*) osErrNoChips=2;
(*36*) osErrNoMem=3;
(*36*) osErrNoChipMem=4;
(*36*) osErrPubNotUnique=5;
(*36*) osErrUnknownMode=6;

(*36*) oScanText=1;
(*36*) oScanStandard=2;
(*36*) oScanMax=3;
(*36*) oScanVideo=4;

```

TYPE

```

(*36*) PubScreenNode =RECORD
    node: Node ;
    screen: ScreenPtr ;
    flags:BITSET;
    size:INTEGER;
    visitorCount:INTEGER;
    sigTask: TaskPtr ;
    sigBit:SHORTCARD;
END;

```

CONST

```

(*
  Bitnummer für PubScreen.flags.
*)
(*36*) psnPrivate=0;

(*36*) maxPubScreenName=139;

(*36*) shanghai=1;
(*36*) popPubScreen=2;

```

CONST

```

filenameSize=30;

pointerSize=(1+16+1)*2;

topazEighty=8;
topazSixty=9;

```

TYPE

```

    BaudRate =(
    baud110,baud300,baud1200,baud2400,baud4800,baud9600,baud19200,baudMidi
    );

    PrintAspect =(aspectHoriz,aspectVert);

    PrintImage =(imagePositive,imageNegative);

```

---

```

PrintQuality =(draft,letter);

PrintShade =(shadeBW,shadeGreyscale,shadeColor);

PrintSpacing =(sixLPI,ps1,eightLPI);

PrinterFlags =(
correctRed,correctGreen,correctBlue,centerImage,boundedDimensions,
absoluteDimensions,pixelDimensions,multiplyDimensions,integerScaling,
halftoneDithering,floydDithering,antiAlias,greyscale2
);
PrinterFlagSet =SET OF  PrinterFlags ;

CONST
correctRGBMask= PrinterFlagSet {correctRed..correctBlue};

dimensionMask= PrinterFlagSet {boundedDimensions..multiplyDimensions};

ditheringMask= PrinterFlagSet {halftoneDithering,floydDithering};

ignoreDimensions= PrinterFlagSet {};

orderedDithering= PrinterFlagSet {};

TYPE
PrinterType =(
customName,alphaP101,brother15XL,cbmMps1000,diab630,diabAdvD25,
diabC150,epson,epsonJX80,okimate20,qumeLP20,hpLaserjet,hpLaserjetPlus
);

PrinterPort =(parallelPrinter,serialPrinter);

SerParShk =(
shakeRts,shakeNone,sps2,sps3,
parityEven,parityOdd,paritySpace,sps7
);
SerParShkSet =SET OF  SerParShk ;

CONST
shakeXon= SerParShkSet {};
parityNone= SerParShkSet {};
(*36*)parityMark= SerParShkSet {parityEven,parityOdd};

TYPE
Preferences =RECORD
fontHeight:SHORTCARD;
printerPort: PrinterPort ;
baudRateMSB:SHORTCARD; (* :=0 *)
baudRate: BaudRate ;
keyRptSpeed: TimeVal ;
keyRptDelay: TimeVal ;
doubleClick: TimeVal ;
pointerMatrix:ARRAY [0..pointerSize-1] OF CARDINAL;
xOffset:SHORTINT;
yOffset:SHORTINT;
color17:CARDINAL;

```

---

```
color18: CARDINAL;
color19: CARDINAL;
pointerTicks: CARDINAL;
color0: CARDINAL;
color1: CARDINAL;
color2: CARDINAL;
color3: CARDINAL;
viewXOffset: SHORTINT;
viewYOffset: SHORTINT;
viewInitX: INTEGER;
viewInitY: INTEGER;
enableCLI: BITSET;
printerTypeMSB: SHORTCARD; (* :=0 *)
printerType: PrinterType ;
printerFilename: ARRAY [0..filenameSize-1] OF CHAR;
printPitch: CARDINAL;
printQuality: PrintQuality ;
printQualityLSB: SHORTCARD; (* :=0 *)
printSpacing: PrintSpacing ;
printSpacingLSB: SHORTCARD; (* :=0 *)
printLeftMargin: CARDINAL;
printRightMargin: CARDINAL;
printImageMSB: SHORTCARD; (* :=0 *)
printImage: PrintImage ;
printAspectMSB: SHORTCARD; (* :=0 *)
printAspect: PrintAspect ;
printShadeMSB: SHORTCARD; (* :=0 *)
printShade: PrintShade ;
printThreshold: INTEGER;
paperSize: CARDINAL;
paperLength: CARDINAL;
paperType: CARDINAL;
serRWBits: SHORTCARD;
serStopBuf: SHORTCARD;
serParShk: SerParShkSet ;
laceWB: SHORTSET;
workName: ARRAY [0..filenameSize-1] OF CHAR;
rowSizeChange: SHORTINT;
columnSizeChange: SHORTINT;
printFlags: PrinterFlagSet ;
printMaxWidth: CARDINAL;
printMaxHeight: CARDINAL;
printDensity: SHORTCARD;
printXOffset: SHORTCARD;
width: CARDINAL;
height: CARDINAL;
depth: SHORTCARD;
size: SHORTCARD;
END;

CONST
pica=0H;
elite=0400H;
fine=0800H;

usLetter=0H;
usLegal=010H;
```

---

```

nTractor=020H;
wTractor=030H;
custom=040H;
(*36*) euroA0=050H;
(*36*) euroA1=060H;
(*36*) euroA2=070H;
(*36*) euroA3=080H;
(*36*) euroA4=090H;
(*36*) euroA5=0A0H;
(*36*) euroA6=0B0H;
(*36*) euroA7=0C0H;
(*36*) euroA8=0D0H;

fanfold=0H;
single=080H;

readBits=0F0H;
writeBits=0FH;

stopBits=0F0H;
bufSizeBits=0FH;

buf512=0;
buf1024=1;
buf2048=2;
buf4096=3;
buf8000=4;
buf16000=5;

(*
  Bitnummer für Preferences.laceWB.
*)
lacewb=0;

(*
  Bitnummerns für Preferences.enableCLI.
*)
(*36*) screenDrag=14;
(*36*) mouseAccel=15;

TYPE
  Remember =RECORD
    nextRemember: RememberPtr ;
    rememberSize: LONGCARD;
    memory: ADDRESS;
  END;

(*36*) ColorSpec =RECORD
  colorIndex: INTEGER;
  red: CARDINAL;
  green: CARDINAL;
  blue: CARDINAL;
END;

(*36*) EasyStruct =RECORD
  structSize: LONGCARD;
  flags: LONGSET;

```

---

```

    title:ADDRESS;
    textFormat:ADDRESS;
    gadgetFormat:ADDRESS;
END;

```

```

TYPE
    DisplayMode =(hiresPick,lowresPick);

```

```

CONST
    dModeCount=ORD (MAX ( DisplayMode ))+1;
    eventMax=10;

```

```

TYPE
    Res =(hiresGadget,lowresGadget);

```

```

CONST
    resCount=ORD (MAX ( Res ))+1;

```

```

TYPE
    Gadgets =(
        upFrontGadget,downBackGadget,sizeGadget,closeGadget,
        dragGadget,sUpFrontGadget,sDownBackGadget,sDragGadget
    );

```

```

CONST
    gadgetCount=ORD (MAX ( Gadgets ))+1;

```

```

TYPE
    IntuitionBase =RECORD
        libNode: Library ;
        viewLord: View ;
        activeWindow: WindowPtr ;
        activeScreen: ScreenPtr ;
        firstScreen: ScreenPtr ;
        flags:LONGSET;
        mouseY:INTEGER;
        mouseX:INTEGER;
        seconds:LONGCARD;
        micros:LONGCARD;
    END;

```

```

TYPE
    (*36*) GadgetInfo =RECORD
        screen: ScreenPtr ;
        window: WindowPtr ;
        requester: RequesterPtr ;
        rastPort: RastPortPtr ;
        layer: LayerPtr ;
        domain: IBox ;
        detailPen:SHORTCARD;
        blockPen:SHORTCARD;
        drInfo: DrawInfoPtr ;
        reserved:ARRAY [0..5] OF LONGCARD;
    END;

```

```

    (*36*) PGX =RECORD

```

---

```
        container: IBox ;
        newKnob: IBox ;
    END;

(*36*) StringExtend =RECORD
    font: TextFontPtr ;
    pens,activePens:ARRAY [0..1] OF SHORTCARD;
    initialModes:LONGSET;
    editHook: HookPtr ;
    workBuffer: StrPtr ;
    reserved:ARRAY [0..3] OF LONGCARD;
END;

(*36*) SGWork =RECORD
    gadget: GadgetPtr ;
    stringInfo: StringInfoPtr ;
    workBuffer: StrPtr ;
    prevBuffer: StrPtr ;
    modes:LONGSET;
    iEvent: InputEventPtr ;
    code:CARDINAL;
    bufferPos:INTEGER;
    numChars:INTEGER;
    actions:LONGSET;
    longInt:LONGINT;
    gadgetInfo: GadgetInfoPtr ;
    editOp:CARDINAL;
END;

CONST
    (*36*) eoNoop=0001H;
    (*36*) eoDelbackward=0002H;
    (*36*) eoDelforward=0003H;
    (*36*) eoMovecursor=0004H;
    (*36*) eoEnter=0005H;
    (*36*) eoReset=0006H;
    (*36*) eoReplacechar=0007H;
    (*36*) eoInsertchar=0008H;
    (*36*) eoBadformat=0009H;
    (*36*) eoBigchange=000AH;
    (*36*) eoUndo=000BH;
    (*36*) eoClear=000CH;
    (*36*) eoSpecial=000DH;

    (*36*) sgmReplace=0;
    (*36*) sgmFixedField=1;
    (*36*) sgmNoFilter=2;
    (*36*) sgmNoChange=3;
    (*36*) sgmNoWorkb=4;
    (*36*) sgmControl=5;
    (*36*) sgmLongint=6;
    (*37*) sgmExitHelp=7;

    (*36*) sgaUse=01H;
    (*36*) sgaEnd=02H;
    (*36*) sgaBeep=04H;
    (*36*) sgaReuse=08H;
```

---



```
(*36*) sgaRedisplay=010H;  
(*37*) sgaNextActive=020H;  
(*37*) sgaPrevActive=040H;
```

```
(*36*) sghKey=1;  
(*36*) sghClick=2;
```

TYPE

```
(*36*) Object =ADDRESS;  
  
(*36*) ClassID = StrPtr ;  
  
(*36*) Msg =POINTER TO RECORD  
        methodID:LONGCARD;  
        END;
```

CONST

```
(*36*) rootClass="rootclass";  
(*36*) imageClass="imageclass";  
(*36*) frameiClass="frameiclass";  
(*36*) sysiClass="sysiclass";  
(*36*) fillrectClass="fillrectclass";  
(*36*) gadgetClass="gadgetclass";  
(*36*) propgClass="propgclass";  
(*36*) strgClass="strgclass";  
(*36*) buttongClass="buttongclass";  
(*36*) frbuttonClass="frbuttononclass";  
(*36*) groupgClass="groupgclass";  
(*36*) icClass="icclass";  
(*36*) modelClass="modelclass";  
  
(*36*) omDummy=0100H;  
(*36*) omNEW=0101H;  
(*36*) omDISPOSE=0102H;  
(*36*) omSET=0103H;  
(*36*) omGET=0104H;  
(*36*) omADDTAIL=0105H;  
(*36*) omREMOVE=0106H;  
(*36*) omNOTIFY=0107H;  
(*36*) omUPDATE=0108H;  
(*36*) omADDMEMBER=0109H;  
(*36*) omREMMEMBER=010AH;
```

TYPE

```
(*36*) OpSet =RECORD  
        methodID:LONGCARD;  
        attrList: TagItemPtr ;  
        gInfo: GadgetInfoPtr ;  
        END;  
  
(*36*) OpUpdate =RECORD  
        methodID:LONGCARD;  
        attrList: TagItemPtr ;  
        gInfo: GadgetInfoPtr ;  
        flags:LONGSET;  
        END;
```

---

CONST

```
(*36*) opufINTERIM=LONGSET{0};
```

TYPE

```
(*36*) OpGet =RECORD
  methodID:LONGCARD;
  attrID:LONGCARD;
  storage:POINTER TO LONGCARD;
END;
```

```
(*36*) OpAddTail =RECORD
  methodID:LONGCARD;
  list: ListPtr ;
END;
```

```
(*36*) OpMember =RECORD
  methodID:LONGCARD;
  object: ObjectPtr ;
END;
```

```
(*36*) OpAddMember = OpMember ;
```

```
(*36*) OpAddMemberPtr = OpMemberPtr ;
```

```
(*36*) IClass =RECORD
  dispatcher: Hook ;
  reserved:LONGCARD;
  super: IClassPtr ;
  id: ClassID ;
  instOffset:CARDINAL;
  instSize:CARDINAL;
  userData:LONGCARD;
  subclassCount:LONGCARD;
  objectCount:LONGCARD;
  flags:LONGSET;
END;
```

CONST

```
(*36*) inList=LONGSET{0};
```

TYPE

(\*

Der Typ \_Object wurde zu UObject umbenannt, weil Modula-2 ein '\_' in einem Namen nicht zulässt, und Object schon verwendet wird.

\*)

```
(*36*) UObject =RECORD
  node: MinNode ;
  class: IClassPtr ;
END;
```

```
(*36*) GaTags =(gaDummy:=tagUser+030000H,
  gaLeft,
  gaRelRight,
  gaTop,
  gaRelBottom,
  gaWidth,
  gaRelWidth,
  gaHeight,
```

```
    gaRelHeight,
    gaText,
    gaImage,
    gaBorder,
    gaSelectRender,
    gaHighlight,
    gaDisabled,
    gaGZZGadget,
    gaID,
    gaUserData,
    gaSpecialInfo,
    gaSelected,
    gaEndGadget,
    gaImmediate,
    gaRelVerify,
    gaFollowMouse,
    gaRightBorder,
    gaLeftBorder,
    gaTopBorder,
    gaBottomBorder,
    gaToggleSelect,
    gaSysGadget,
    gaSysGType,
    gaPrevious,
    gaNext,
    gaDrawInfo,
    gaIntuiText,
    gaLabelImage,
    gaTabCycle
);

(*36*) PgaTags = (pgaDummy:=tagUser+031000H,
    pgaFreedom,
    pgaBorderless,
    pgaHorizPot,
    pgaHorizBody,
    pgaVertPot,
    pgaVertBody,
    pgaTotal,
    pgaVisible,
    pgaTop,
    (*37*) pgaNewLook
);

(*36*) StringaTags = (stringaDummy:=tagUser+032000H,
    stringaMaxChars,
    stringaBuffer,
    stringaUndoBuffer,
    stringaWorkBuffer,
    stringaBufferPos,
    stringaDispPos,
    stringaAltKeyMap,
    stringaFont,
    stringaPens,
    stringaActivePens,
    stringaEditHook,
    stringaEditModes,
```

---

```

        stringaReplaceMode,
        stringaFixedFieldMode,
        stringaNoFilterMode,
        stringaJustification,
        stringaLongVal,
        stringaTextVal,
        (*37*) stringaExitHelp
    );

```

CONST

```

    (*36*) sgDefaultMaxChars=128;

```

TYPE

```

    (*36*) LayoutaTags =(layoutaDummy:=tagUser+038000H,
        layoutaLayoutObj,
        layoutaSpacing,
        layoutaOrientation
    );

```

CONST

```

    (*36*) lorientNone=0;
    (*36*) lorientHoriz=1;
    (*36*) lorientVert=2;

```

```

    (*36*) gmDummy=-1;
    (*36*) gmHITTEST=0;
    (*36*) gmRENDER=1;
    (*36*) gmGOACTIVE=2;
    (*36*) gmHANDLEINPUT=3;
    (*36*) gmGOINACTIVE=4;

```

TYPE

```

    (*36*) GpHitTest =RECORD
        methodID:LONGCARD;
        gInfo: GadgetInfoPtr ;
        mouse: Point ;
    END;

```

CONST

```

    (*36*) gmrGadgetHit=000000004H;

```

TYPE

```

    (*36*) GpRender =RECORD
        methodID:LONGCARD;
        gInfo: GadgetInfoPtr ;
        rPort: RastPortPtr ;
        redraw:LONGINT;
    END;

```

CONST

```

    (*36*) gRedrawUpdate=2;
    (*36*) gRedrawRedraw=1;
    (*36*) gRedrawToggle=0;

```

TYPE

```

    (*36*) GpInput =RECORD
        methodID:LONGCARD;

```

```

    gInfo: GadgetInfoPtr ;
    iEvent: InputEventPtr ;
    termination:ADDRESS;
    mouse: Point ;
END;

```

```

(*36*) GpGoInactive =RECORD
    methodID:LONGCARD;
    gInfo: GadgetInfoPtr ;
    (*37*) abort:LONGCARD;
END;

```

CONST

```

(*)
gmrXXXX sind Bitnummern eines LONGSETs!
*)
(*36*) gmrMeActive=0;
(*36*) gmrNoReuse=1;
(*36*) gmrReuse=2;
(*36*) gmrVerify=3;
(*37*) gmrNextActive=4;
(*37*) gmrPrevActive=5;

(*36*) icmDummy=0401H;
(*36*) icmSETLOOP=0402H;
(*36*) icmCLEARLOOP=0403H;
(*36*) icmCHECKLOOP=0404H;

```

TYPE

```

(*36*) IcaTags =(icaDummy:=tagUser+040000H,
    icaTarget,
    icaMap,
    icspecialCode
);

```

CONST

```

(*36*) icTargetIDCMP=-1;

(*36*) customImageDepth=-1;

```

TYPE

```

(*36*) IaTags =(iaDummy:=tagUser+020000H,
    iaLeft,
    iaTop,
    iaWidth,
    iaHeight,
    iaFGPen,
    iaBGPen,
    iaData,
    iaLineWidth,
    iaShadowPen,
    iaHighlightPen,
    sysiaSize,
    sysiaDepth,
    sysiaWhich,
    iaPens,
    iaResolution,

```

```

        iaAPattern,
        iaAPatSize,
        iaMode,
        iaFont,
        iaOutline,
        iaRecessed,
        iaDoubleEmboss,
        iaEdgesOnly,
        sysiaDrawInfo
    );

```

CONST

```

    (*36*) sysiaPens=iaPens;

```

```

    (*36*) sysisizeMedres=0;
    (*36*) sysisizeLowres=1;
    (*36*) sysisizeHires= 2;

```

```

    (*36*) depthImage=00H;
    (*36*) zoomImage=01H;
    (*36*) sizeImage=02H;
    (*36*) closeImage=03H;
    (*36*) sdepthImage=05H;
    (*36*) leftImage=0AH;
    (*36*) upImage=0BH;
    (*36*) rightImage=0CH;
    (*36*) downImage=0DH;
    (*36*) checkImage=0EH;
    (*36*) mxImage=0FH;

```

```

    (*36*) imDraw=202H;
    (*36*) imHitTest=203H;
    (*36*) imErase=204H;
    (*36*) imMove=205H;
    (*36*) imDrawFrame=206H;
    (*36*) imFrameBox=207H;
    (*36*) imHitFrame=208H;
    (*36*) imEraseFrame=209H;

```

```

    (*36*) idsNormal=0;
    (*36*) idsSelected=1;
    (*36*) idsDisabled=2;
    (*36*) idsBusy=3;
    (*36*) idsIndeterminate=4;
    (*36*) idsInactiveNormal=5;
    (*36*) idsInactiveSelected=6;
    (*36*) idsInactiveDisabled=7;

```

TYPE

```

    (*36*) ImpFrameBox =RECORD
        methodID:LONGCARD;
        contentsBox: IBoxPtr ;
        frameBox: IBoxPtr ;
        drInfo: DrawInfoPtr ;
        frameFlags:LONGSET;
    END;

```

```
CONST
  (*36*) framefSpecify=LONGSET{0};
```

```
TYPE
  (*36*) ImpDraw =RECORD
    methodID:LONGCARD;
    rPort: RastPortPtr ;
    offset:RECORD
      x:INTEGER;
      y:INTEGER;
    END;
    state:LONGCARD;
    drInfo: DrawInfoPtr ;
    dimensions:RECORD
      width:INTEGER;
      height:INTEGER;
    END;
  END;
```

```
  (*36*) ImpErase =RECORD
    methodID:LONGCARD;
    rPort: RastPortPtr ;
    offset:RECORD
      x:INTEGER;
      y:INTEGER;
    END;
    dimensions:RECORD
      width:INTEGER;
      height:INTEGER;
    END;
  END;
```

```
  (*36*) ImpHitTest =RECORD
    methodID:LONGCARD;
    point: Point ;
    dimensions:RECORD
      width:INTEGER;
      height:INTEGER;
    END;
  END;
```

```
END IntuitionD.noimp
```

---