

RexxD

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RexxD	1
1.1	RexxD	1
1.2	TMP:Modula-2/RexxD.def	1

Chapter 1

RexxD

1.1 RexxD

Konstanten

alpha	append	buffsz
dpNum	exist	globalsz
intNum	keepNum	keepStr
maxRMArg	memMask	memQuant
memQuick	owned	queue
read	rexName	rxschunk
rxsdir	rxsname	rxsnest
rxsstack	rxstname	rxstpri
stack	stop	strf
write		

Typ-Deklarationen

ActionRec	Argstring	AttributeFlagSet
AttributeFlags	CharAttrFlagSet	CharAttrFlags
Command	IoBuff	IoBuffPtr
ModifierFlagSet	ModifierFlags	NexxStr
NexxStrPtr	RexxArg	RexxArgPtr
RexxMsg	RexxMsgPort	RexxMsgPtr
RexxRsrc	RexxRsrcPtr	RexxTask
RexxTaskFlagSet	RexxTaskFlags	RexxTaskPtr
RsrcNodeType	RxsLib	RxsLibPtr
SrcNode	SrcNodePtr	StrPtr

1.2 TMP:Modula-2/RexxD.def

```
DEFINITION MODULE RexxD; (*$Implementation:=FALSE*)
(* 15-May-1992/cn *)
```

```
FROM SYSTEM IMPORT ADDRESS, BPTR;
```

```
FROM DosD IMPORT
```

```

    DosLibraryPtr , FileHandlePtr , FileLockPtr ;

FROM ExecD IMPORT
    ExecBasePtr , Library , List , ListPtr , MemReqs , Message , MessagePtr
    , MsgPort , MsgPortPtr , Node ;

CONST
    rexxName="rexsyslib.library";
    rxpname=rexName;
    rxmdir="REXX";
    rxstname="ARexx";

TYPE
    Argstring  = ADDRESS;

    StrPtr =ADDRESS;

    AttributeFlags =(keep,string,notNum,number,binary,float,ext,source);
    AttributeFlagSet =SET OF  AttributeFlags ;

    NexxStr =RECORD
        iValue:LONGINT;
        length:CARDINAL;
        flags: AttributeFlagSet ;
        hash:SHORTCARD;
        (* buff:ARRAY OF CHAR; *)
    END;
    NexxStrPtr =POINTER TO  NexxStr ;

CONST
    intNum= AttributeFlagSet {number,binary,string};
    dpNum= AttributeFlagSet {number,float};
    alpha= AttributeFlagSet {notNum,string};
    owned= AttributeFlagSet {source,ext,keep};
    keepStr= AttributeFlagSet {string,source,notNum};
    keepNum= AttributeFlagSet {string,source,number,binary};

TYPE
    RexxArg =RECORD
        size:LONGINT;
        length:CARDINAL;
        flags: AttributeFlagSet ;
        hash:SHORTCARD;
        (* buff:ARRAY OF CHAR; *)
    END;
    RexxArgPtr =POINTER TO  RexxArg ;

CONST
    maxRMArg=15;

TYPE
    Command =(
        co0,comm,func,close,query,co5,co6,addFH,
        addLib,remLib,addCon,remCon,tcOpn,tcCls
    );
    ModifierFlags =(

```

```

    noIO,result,moString,token,nonRet,mo5,mo6,mo7
);
ModifierFlagSet =SET OF  ModifierFlags ;

ActionRec =RECORD
command: Command ;
modifier: ModifierFlagSet ;
add:CARDINAL;
END;

RexxMsg =RECORD
node: Message ;
taskBlock:ADDRESS;
libBase:ADDRESS;
action: ActionRec ;
result1:LONGINT;
result2:LONGINT;
args:ARRAY [0..maxRMArg] OF  StrPtr ;
passPort: MsgPortPtr ;
commAddr: StrPtr ;
fileExt: StrPtr ;
stdin: FileHandlePtr ;
stdout: FileHandlePtr ;
avail:LONGINT;
END;
RexxMsgPtr =POINTER TO  RexxMsg ;

RexxRsrc =RECORD
node: Node ;
func:INTEGER;
base:ADDRESS;
size:LONGINT;
arg1:LONGINT;
arg2:LONGINT;
END;
RexxRsrcPtr =POINTER TO  RexxRsrc ;

RsrcNodeType =(any,lib,port,file,host,clip);

CONST
globalsz=200;

TYPE
RexxTaskFlags =(trace, halt, susp, tcUse, rtf4, rtf5, wait, rtfClose);
RexxTaskFlagSet =SET OF  RexxTaskFlags ;

RexxTask =RECORD
global:ARRAY [0..globalsz-1] OF SHORTINT;
msgPort: MsgPort ;
flags: RexxTaskFlagSet ;
sigBit:SHORTINT;
clientID:ADDRESS;
msgPkt:ADDRESS;
taskID:ADDRESS;
rexPort:ADDRESS;
errTrap:ADDRESS;
stackPtr:ADDRESS;

```

```
envList: List ;
freeList: List ;
memList: List ;
fileList: List ;
portList: List ;
END;
RexxTaskPtr =POINTER TO  RexxTask ;

CONST
memQuant=16;
memMask=FFFFFFFF0H;
memQuick=public;

TYPE
SrcNodePtr =POINTER TO  SrcNode ;
SrcNode =RECORD
succ: SrcNodePtr ;
pred: SrcNodePtr ;
ptr:ADDRESS;
size:LONGINT;
END;

TYPE
RxsLib =RECORD
node: Library ;
flags: RexxTaskFlagSet ;
shadow:SHORTCARD;
sysBase: ExecBasePtr ;
dosBase: DosLibraryPtr ;
ieeeDPBase:ADDRESS;
segList:BPTR;
nil: FileHandlePtr ;
chunk:LONGINT;
maxNest:LONGINT;
null: NexxStrPtr ;
false: NexxStrPtr ;
true: NexxStrPtr ;
rex: NexxStrPtr ;
command: NexxStrPtr ;
stdin: NexxStrPtr ;
stdout: NexxStrPtr ;
stderr: NexxStrPtr ;
version: StrPtr ;
taskName: StrPtr ;
taskPri:LONGINT;
taskSeg:BPTR;
stackSize:LONGINT;
rexDir: StrPtr ;
cTable: StrPtr ;
notice: StrPtr ;
rexPort: MsgPort ;
readLock:CARDINAL;
traceFH: FileHandlePtr ;
taskList: List ;
numTask:INTEGER;
libList: List ;
numLib:INTEGER;
```

```
clipList: List ;
numClip: INTEGER;
msgList: List ;
numMsg: INTEGER;
pgmList: List ;
numPgm: INTEGER;
traceCnt: CARDINAL;
avail: INTEGER;
END;
RxsLibPtr = POINTER TO RxsLib ;

CONST
stop=wait;

rxschunk=1024;
rxsnest=32;
rxstpri=0;
rxsstack=4096;

TYPE
CharAttrFlags =(
space, digit, caAlpha, rexxSym, rexxOpr, rexxSpc, upper, lower
);
CharAttrFlagSet = SET OF CharAttrFlags ;

CONST
buffsz=204;

TYPE
IoBuff = RECORD
node: RxxRsrc ;
rpt: ADDRESS;
rct: LONGINT;
dfh: FileHandlePtr ;
lock: FileLockPtr ;
bct: LONGINT;
area: ARRAY [0..buffsz-1] OF SHORTINT;
END;
IoBuffPtr = POINTER TO IoBuff ;

CONST
exist=-1;
strf=0;
read=1;
write=2;
append=3;

TYPE
RxxMsgPort = RECORD
node: RxxRsrc ;
port: MsgPort ;
replyList: List ;
END;

CONST
stack=2002;
queue=2003;
```

END RexxD.noimp