

JWlink Version 1.9

Portions Copyright (c) 1985-2002 Sybase, Inc. All Rights Reserved.

Source code is available under the Sybase Open Watcom Public License.

usage: jwlink {directive}

commands valid for all executable formats:

```
directive ::= "File"          obj_spec{",","obj_spec"}
           | "Name"          exe_file
           | "OPTion"        option{",","option"}
           | "Library"       library_file{",","library_file"}
           | "OPTLIB"        library_file{",","library_file"}
           | "Path"          path_name{";","path_name"}
           | "LIBPath"       path_name{";","path_name"}
           | "LIBFile"       obj_file{",","obj_file"}
           | "Debug"         dbtype [dblist] | dblist
           | "MODTrace"      module_name{",","module_name"}
           | "SYMTrace"      symbol_name{",","symbol_name"}
           | "SYSTEM"        system_name
           | "SYStem Begin"  system_name directive {directive} "End"
           | "FORMat"        form
           | "REFerence"     symbol_name{",","symbol_name"}
           | "@"             directive_file
           | "#"             comment
```

commands valid for all executable formats (continued):

```
           | "Alias"         alias="symbol_name{",","alias"}="symbol_name}
           | "DISAbLe"       msg_num{",","msg_num"}
           | "SOrt"          ["GLobal"] ["ALPhabetical"]
           | "LANGUAge"      ("JApAnese" | "CHinese" | "KOrean")
           | "STARTLINK"
           | "ENdLINK"
           | "MODFile"       obj_file{",","obj_file"}
obj_spec  ::= obj_file["("obj_member)"] | library_file["("lib_member)"]
dbtype    ::= "Watcom" | "Dwarf" | "Codeview"
dblist    ::= dboption{",","dboption"}
dboption  ::= "LInes" | "Types" | "LLocals" | "All"
option    ::= "ARTificial" | "[NO]CACHe" | "[NO]Caseexact" | "CVPack"
           | "Dosseg" | "ELIminate" | "[NO]FARcalls"
           | "INCremental[=file_name]" | "MANGlednames"
           | "Map"[="map_file"] | "MAXErrors="n | "NAMELen="n
           | "NODefaultlibs" | "NOEXtension" | "OSName="string | "Quiet"
           | "[NO]REDefsoK" | "STack="n | "START="symbol_name | "STATics"
           | "SYMFile"[="symbol_file"], | "[NO]UndefsoK" | "VerboSe"
           | "VFRemoval"
```

commands valid for the MS-DOS executable format only:

```
form      ::= "Dos" ["COM"]
directive ::= "NEWsegment"
option    ::= "PACKCode="n | "PACKData="n | KNOWEAS
```

commands valid for the OS/2 executable formats (NE, LE, LX) only:

```
form      ::= "OS2" ["FLat"|"LE"|"LX"] ["PHYSdevice" | "VIRTdevice"
           | ["DL1"|"INITGlobal"|"INITInstance" ["TERMInstance"|"TERMGlobal"]]
           | "PM" | "PMCompatible" | "FULLScreen"]
directive ::= "NEWsegment"
           | "SEGment"      segdesc{",","segdesc"}
           | "IMPort"       import{",","import"}
           | "EXPort"       export{",","export"}
           | "EXPort"       "wlib_directive_file"
segdesc   ::= segspec segmodel {segmodel}
segspec   ::= seg_name | "Class" class_name | "TYpe" ("CODE" | "DATA")
import    ::= entryname module["."(extname | ordinal)]
option    ::= "Alignment="n | "DEscription" description | "Heapsize="n
           | "IMPFile"[=file_name] | "IMPLib"[=file_name]
           | ("MANYautodata" | "NOautodata" | "ONEAutodata")
```

```

    | "MODName="module_name | "MIXed1632" | "NEWFiles" | "OFFset="n
    | "OLDlibrary="dll_name | "PACKCode="n | "PACKData="n | "PROTmode"
    | "RESource="res_file_name | "STUB="stub_name | "NOSTUB"
    | "VERSion="major.[minor]
export ::= entryname["."ordinal][ "="internal][ "RESident"][ "PRIVATE"][ iopl]
segmodel ::= ("Preload" | "LOadoncall") | ("SHared" | "NONShared")
    | ("EXECUTEOnly" | "EXECUTERead" | "READOnly" | "READWrite")
    | ("Iopl" | "NOIopl") | ("CONforming" | "NONConforming")
    | ("PERmanent" | "NONPERmanent") | "INValid" | "RESident"
    | "CONTiguous" | "DYNamic"

```

commands valid for the Microsoft Win16 executable formats (NE) only:

```

form ::= "WIndows" ["DLl"]["INITGlobal" | "INITInstance"] [MEMory] [FOnt]
directive ::= "NEWsegment"
    | "SEGment" segdesc{","segdesc}
    | "IMPort" import{","import}
    | "EXPort" export{","export}
    | "EXPort" "="wlib_directive_file
segdesc ::= segspec segmodel {segmodel}
segspec ::= seg_name | "Class" class_name | "TYpe" ("CODE" | "DATA")
import ::= entryname module["."(extname | ordinal)]
option ::= "Alignment="n | "DEscription" description | "Heapsize="n
    | "IMPFile" [=file_name] | "IMPLib" [=file_name]
    | ("MANYautodata" | "NOautodata" | "ONEAutodata")
    | "MODName="module_name | "OLDlibrary="dll_name | "PACKCode="n
    | "PACKData="n | "RESource="res_file_name | "RWReloccheck"
    | "STUB="stub_name | "NOSTUB" | "VERSion="major.[minor]
export ::= entryname["."ordinal][ "="internal][ "RESident"][ "PRIVATE"]
segmodel ::= ("Preload" | "LOadoncall") | ("SHared" | "NONShared")
    | ("EXECUTEOnly" | "EXECUTERead" | "READOnly" | "READWrite")
    | ("MOVEable" | "FIXed") | "DIScardable"

```

commands valid for the Microsoft Windows VxD formats only:

```

form ::= "WIndows VXD" ["DYNamic"]

```

commands valid for the PE executable formats only:

```

form ::= "WIndows PE" ["TNT"|"HX"]
    ["DLl"]["INITGlobal"|"INITInstance" ["TERMinstance"|"TERMGlobal"]]
directive ::= "RUntime" runoption["="major["."minor]]
    | "SEGment" segdesc{","segdesc}
    | "IMPort" import{","import}
    | "EXPort" export{","export}
    | "EXPort" "="wlib_directive_file
    | "COMmit" commitopt{","commitopt}
segdesc ::= segspec segmodel {segmodel}
segspec ::= seg_name | "Class" class_name | "TYpe" ("CODE" | "DATA")
import ::= entryname module["."(extname | ordinal)]
option ::= "Alignment="n | "DEscription" description | "Heapsize="n
    | "IMPFile" [=file_name] | "IMPLib" [=file_name]
    | "MODName="module_name | "NOSTDCall" | "OBJAlign="n | "OFFset="n
    | "OLDlibrary="dll_name | "RESource="res_file_name
    | "STUB="stub_name | "NOSTUB" | "VERSion="major.[minor]
    | "[NO]LARGEaddressaware
export ::= entryname["."ordinal][ "="internal][ "PRIVATE"]
segmodel ::= ("SHared" | "NONShared") | ("PAGEable" | "NONPageable")
    | "EXECUTable" | "WRITable"
runoption ::= ("NATive" | "WIndows" | "CONsole" | "POSix" | "OS2" | "DOSstyle")
commitopt ::= "STack="n | "Heap="n

```

commands valid for the Phar Lap executable formats only:

```

form ::= "PHARlap" ["EXTended" | "REX" | "SEGmented"]
directive ::= "RUntime" runoption{","runoption}

```

```

option      ::= "MAXData="n | "MINData="n | "OFFset="n | "STUB="stub_name
runoption   ::= "MINReal="n | "MAXReal="n | "CALLBufs="n | "MINIBuf="n
               | "MAXIBuf="n | "NISTack="n | "ISTKsize="n | "PRIVileged"
               | "REALBreak="(n | symbol_name ) | "UNPRIVileged"

```

commands valid for the Novell Netware executable format only:

```

form        ::= "NOVell" ["NLM" | "LAN" | "DSK" | "NAM" | 'number'] 'description'
directive   ::= "MODUle"      module_name{"", "module_name"}
               | "IMPorT"     import{"", "import"}
               | "EXPorT"     export{"", "export"}
               | "Debug"      "Novell" ["ONLYexports"] ["REFerenced"]
option      ::= "ChECk="symbol_name | "COpyRight" [string] | "CUSTom="file_name
               | "EXit="symbol_name | "HElp="file_name | "IMPFile="file_name
               | "IMPLib"[=file_name] | "MESsages"[=file_name] | "MULTILoad"
               | "NLMFlags="value | "PSeudopreemption" | "REentrant"
               | "SCReenname" description | "SHARelib="file_name | "SYnchronize"
               | "THReadname" name | "VERsion="major[.minor[.revision]]
               | "XDCdata="file_name | "AUTOUNload"
import      ::= entryname
export      ::= entryname

```

commands valid for the QNX executable format only:

```

form        ::= "QNX" ["FLat"]
directive   ::= "SEGment"     segdesc{"", "segdesc"}
               | "NEwsegment"
option      ::= "Heapsize="n | "LInearrelocs" | "LONglived" | "NORelocs"
               | "OFFset="n | "PACKCode="n | "PACKData="n | "PRIVilege"=n
               | "RESource"("="res_file_name | string)
segdesc     ::= segspec segmodel {segmodel}
segspec     ::= seg_name | "Class" class_name | "TYpe" ("CODE" | "DATA")
segmodel    ::= "EXECUTEOnly" | "EXECUTERead" | "READOnly" | "READWrite"

```

commands valid for the ELF executable format only:

```

form        ::= "ELF" ["DL1"]
directive   ::= "MODUle"      module_name{"", "module_name"}
               | "IMPorT"     import{"", "import"}
               | "EXPorT"     export{"", "export"}
               | "RUntime"     runoption{"", "runoption"}
option      ::= "AligNment="n | "EXPorTAll" | "EXTRASections="n | "OBJAligN="n
               | "OFFSet="n
import      ::= entryname
export      ::= entryname
runoption   ::= ("SVR4" | "LINUX" | "FREebSD" | "NETbsd" | "SOLaris" | "ABIver" )

```