EXJLD(8)

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

SYNOPSIS

exjld [*OPTIONS*] JAR...

Chapter 2

DESCRIPTION

exjld is Enduro/X Java linker too. The tool is used to compile given jar files into single executable binary. **exjld** basically extracts all jar files, converts them to C resources (statically initialized byte arrays). These byte array global variables are indexed by class names. The application then is started by custom class loader which from JNI side pulls is the classes necessary. Basically by using this tool, the developer can distributed Java programs with zero dependencies, except the fact that JDK and Enduro/X libraries must be installed on system. All other depended JARs are compiled statically into single executable. This can greatly help to improve product development and streamline the product delivery to the customer. As the all logic is encapsulated into autonomous binary. Thus at the production, the target system can be composed of different such kind of binaries, where each of them can depend on different set or version of the third party Jar libs.

exjld linker normally links the binary two times. First time it does the test when class loader is instructed to load main class. If loading succeed, then linker re-links the binary with the final mode which includes the execution of the main class after it is loader.

exjld must have access to **cc** (C Compiler). **exjld** by default calls Enduro/X **buildserver** utility used for compiling sources and linking with Enduro/X libs. Also program uses **jar** tool for extracting the jar files. The class files to C resources are converted by **exembedfile(8)** utility.

Chapter 3

ENVIRONMENT

CC
Optional environment variable indicating which C compiler to use. If parameter is not set, the **cc** command is used.

CFLAGS
Optional C flags to be passed to C compiler during the **buildserver** execution.

NDRX_HOME
Optional Enduro/X home directory, where to search for the **include** sub-folder and library folder.

Chapter 4

OPTIONS

JAR
This is list of JAR files that must be linked into binary.

-m ***MAIN_CLASS_NAME***
This is main class name into which executable will enter after the process is started.

[-o ***OUTPUT_BINARY*****]**
This is name of output binary.If not specified then **a.out** is used.

[-l ***LINK_LIBRARIES*****]**
Additional shared/static C libraries that shall be linked with the process. The *-l* argument can be repeated multiple times. The list of libraries are passed down to the **buildserver** (or alternative) tool into format of C style libs string.

[-L 'LIBRARY_SEARCH_PATH]
This is link time (and also used for test runtime) the path where to search for libraries used by compiler. Either it is path of static or dynamic libraries. Can be repeated multiple times.

[-I ***INCLUDE_PATH*****]**
This is path where C compiler shall search for C include headers. Can be repeated multiple times.

[-b ***BUILD_COMMAND*****]**
This is override of the C build command, which by default is Enduro/X's **buildserver**.

[-n]
Do not perform test run of the main class loader.

[-t ***TEMP_DIR_PREFIX*****]**
The default prefix of temp dir (where JARs are extracted) is *./*. With this parameter other directory can be chosen.

[-e ***EMBED_RESOURCES*****]**
Additional resource files that can be accessible via Enduro/X Java API.

[-j ***NRTHREADS*****]**
Number of threads used for embedded code generation. Default is **4**.

[-h]
Print usage.

Chapter 5

EXIT STATUS

0
Success

1
Failure

Chapter 6

EXAMPLE

See **tests/00_unit/CMakeLists.txt** for sample usage.

Chapter 7

BUGS

Report bugs to support@mavimax.com

Chapter 8

SEE ALSO

buildserver(8)

Chapter 9

COPYING