NDRXDEBUG.CONF(5)

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

SYNOPSIS

BINARY_NAME [ndrx=NDRX_DEBUG_LEVEL] [ubf=UBF_DEBUG_LEVEL] [tp=TP_DEBUG_LEVEL] [bufsz=DEBUG_BUFFE
[threaded=THREADED] [mkdir=MKDIR] file=[LOG_FILE] [iflags=INTEGRATION_FLAGS]

Chapter 2

DESCRIPTION

debug.conf a file specified by *NDRX_DEBUG_CONF* env variable configures EnduroX platform logging for each of the separate executable. The logging is done from two sources. One source is UBF and another is ATMI (ndrx). For each of the sources debug level can be set for each separate binary. Debug file or stderr also can be specified for each of the separate binaries.

Chapter 3

DEBUG CONFIGURATION FILE SYNTAX

BINARY_NAME
> Binary name is either executable name or *. In case of * it is applied to default configuration. It is recommended to put it in start of the file. Any binary afterwards in file will override defaults.

NDRX_DEBUG_LEVEL
> ATMI debug level. 0..5. 0=No logging. 1=Fatal, 2=Error, 3=Warning, 4=Program info, 5=Program detail.

UBF_DEBUG_LEVEL
> UBF debug level. 0..5. 0=No logging. 1=Fatal, 2=Error, 3=Warning, 4=Program info, 5=Program detail.

TP_DEBUG_LEVEL
> This is tplog(3) user logging level. 0..5. 0=No loging. 1=Fatal, 2=Error, 3=Warning, 4=Program info, 5=Program detail.

DEBUG_BUFFER_SIZE
> Number of lines to buffer before writing out to disk.

THREADED
> Value is can be set to "Y" or "N". The default is "N". In case of "N" then logging of the process debug output is sent to the *LOG_FILE*. All threads are doing logging to single file. If set to "Y", then for each XATMI client there will be open new log file. The log file names are assigned by following approach: Say *LOG_FILE* = **TEST.LOG**, then new thread log will go to **TEST.X.LOG** (.e.g **TEST.1.LOG**). If *LOG_FILE* = **TEST_LOG**, then new thread will perform logging to "TEST_LOG.1".

MKDIR
> Value is can be set to "Y" or "N". The default is "N". In case of "Y", if Enduro/X logger attempts to open log file and the output directory is missing i.e. fopen() gives **ENOENT** error, the logger will attempt to create the missing folders below recursively. This functionality might be helpful if system is running in active-active mode with out shared disk volume and using request file logging in different folders. Thus if call is routed to other cluster node, it can create the exact file name locally with performing log file switching.

LOG_FILE
> Log file. If empty then *stderr* will be used.

COMMENTS
> Commented lines starts with *#*. Empty lines are ignored.

INTEGRATION_FLAGS
> Integration specific flags. These specific for binding environments. For example **endurox-go** package have defined flag *detailed* which will print the Go source file name and the code line instead of fixed C function file/line.

Chapter 4

EXAMPLE

Sample configuration:

```
# Global config:
* ndrx=5 ubf=1 lines=1 bufsz=1000 tp=5 file=
# Per binary config:
xadmin          file=${NDRX_APPHOME}/tmp/XADMIN
ndrxd           file=${NDRX_APPHOME}/tmp/NDRXD
myclient        file=${NDRX_APPHOME}/tmp/MYCLIENT
myserver        file=${NDRX_APPHOME}/tmp/MYSERVER threaded=y
ud              file=${NDRX_APPHOME}/tmp/NDRX
```

Chapter 5

COMMON CONFIGURATION INI SECTION

Similar syntax is used to describe the debug configuration when common configuration is used (i.e. configuration in ini files). The difference is that there is one extra equation mark between the binary and debug string, see example:

```
[@debug]
# Global config:
*=ndrx=5 ubf=1 lines=1 bufsz=1000 tp=5 file=
# Per binary config:
xadmin=file=${NDRX_APPHOME}/tmp/XADMIN
ndrxd=file=${NDRX_APPHOME}/tmp/NDRXD
myclient=file=${NDRX_APPHOME}/tmp/MYCLIENT
myserver=file=${NDRX_APPHOME}/tmp/MYSERVER threaded=y
ud=file=${NDRX_APPHOME}/tmp/NDRX
```

Chapter 6

BUGS

Report bugs to support@mavimax.com

Chapter 7

SEE ALSO

xadmin(8), ndrxd(8), ndrxconfig.xml(5)

Chapter 8

COPYING