

**EX\_ENV(5)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>DESCRIPTION</b>	<b>1</b>
<b>2</b>	<b>Mandatory Env. Variables</b>	<b>2</b>
<b>3</b>	<b>EXAMPLE</b>	<b>7</b>
<b>4</b>	<b>NOTES FOR ORACLE DB</b>	<b>10</b>
<b>5</b>	<b>BUGS</b>	<b>11</b>
<b>6</b>	<b>SEE ALSO</b>	<b>12</b>
<b>7</b>	<b>COPYING</b>	<b>13</b>

## Chapter 1

# DESCRIPTION

Enduro/X use set of environment variables which is initial environment for application domain. There is set of parameters related to system, prefixes, Posix Queue mount points, various system limits. Next step to the configuration is application domain configuration by it self, in *ndrxconfig.xml* and *debug.conf* (or Common-Configuration respectively).

---

## Chapter 2

# Mandatory Env. Variables

**NDRX\_NODEID=NODE\_ID**

Cluster Node ID. Should be within 1..32.

**NDRX\_CLUSTERISED=1|0**

Do we work in cluster mode or not? *1* - yes, we are in cluster. *0* - no, we are not in cluster.

**NDRX\_LDBAL=LOAD\_BALANCE\_PERCENT**

if `LOAD_BALANCE_PERCENT` is 0 = means process all locally, if value is set to 100 = process all on remote servers (if svc available on remote server). 70% means, process 30% locally 70% remotely.

**NDRX\_TOUT=TIME\_OUT\_IN\_SECONDS**

Global `tpcall()` timeout, if not called with `TPNOTIME` flag. If process will not return in this time, then `TPETIME` error will be returned.

**NDRX\_ULOG=ULOG\_PATH**

Writable system path where to write user logs.

**NDRX\_QPREFIX=APP\_PREFIX**

Prefix used for POSIX IPC resources (shm and queue).

**NDRX\_SVCMAX=MAX\_SERVICES**

This determines the maximum number of services supported by instance. The less is number, then less shared memory is used. However for performance reason, number should be much bigger (3-4 times) than actual estimated service count. Bigger number might speed up linear hashing, i.e. it will be less likely that hash will collide.

**NDRX\_SRVMAX=MAX\_SERVERS**

Maximum number of servers that will be supported. This affects the `-i` flag. Server ID. The max server id will be `MAX_SERVERS-1`.

**NDRX\_CONFIG=FULL\_PATH\_TO\_CONFIG\_FILE**

This shows the full path to usual `ndrxconfig.xml`.

**NDRX\_QPATH=QUEUE\_MOUNT\_POINT**

Full path to directory where POSIX Queue is mounted.

**NDRX\_SHMPATH=SHARED\_MEM\_MOUNT\_POINT**

Full path to POSIX Shared memory mount point.

**NDRX\_CMDWAIT=NDRXD\_COMMAND\_WAIT\_SECS**

Number of second to wait for commands, after which `ndrxd` does other checks.

**NDRX\_DPID=PATH\_TO\_NDRXD\_PID\_FILE**

This is full path to `ndrxd` PID file.

---

**NDRX\_RNDK=APP\_DOMAIN\_RANDOM\_KEY**

Random key to indentify the processes belonging to this app domain (i.e. used in ps ef).

**NDRX\_IPCKEY=SYS\_V\_SEMAPHORE\_KEY**

System V Semaphores key number. Used for shared memory locking. POSIX semaphores are not used here, because they are not unlocked if process dies during acquired semaphore.

**NDRX\_MSGMAX=MAX\_MSGS\_PER\_QUEUE**

Posix queue config attrib. Max number of messages that can be put in one queue.

**NDRX\_DQMAX=NDRXD\_ACCESS\_Q\_MSG\_MAX**

Enduro/X ATMI daemoni *ndrxd* access Q size. Bigger is better.

**NDRX\_MSGSIZEMAX=NDRXD\_ACCESS\_Q\_MSG\_MAX**

Max message size (in bytes). The value is limited to the operating system limits of the maximums size of the POSIX queues message size. For example on Linux kernel 3.13 it is possible to set message size to 10 Megabytes. Also the message size denotes the internal buffer sizes for XATMI message processing. As mostly for all operations Enduro/X uses stack allocation for the buffers, this means that there are requirements for the application stack sizes. Thus the stack size must be 30 times the size of the message. For example if message size is set to 64KB, then stack size must be at least 1920KB (~2MB). Otherwise application will not start. If message size max is set to less than 64KB, then buffer size is defaulted to 64KB. This parameter also affects the opening/creation of the message queue. As the message size is specified in *mq\_open()* attributes.

**NDRX\_APPHOME=FULL\_PATH\_TO\_APPDOMAIN\_INSTANCE\_DIR**

This is full path to application (not an Enduro/X directory it self) root directory.

**NDRX\_HOME=FULL\_PATH\_TO\_ENDUROX\_ROOT\_DIRECTORY**

Full path to Enduro/X installation directory.

**NDRX\_DEBUG\_CONF=FULL\_PATH\_TO\_DEBUG\_CONFIG\_FILE**

This sets full path to debug configuration file.

**NDRX\_UBFMAXFLDS=MAX\_NUMBER\_OF\_UBFFIELDS**

Max number of UBF fields. Used for hashing. Bigger number is better. The max number is number is 33554432 (25 bit).

**NDRX\_DMNLOG=FULL\_PATH\_TO\_NDRX\_DMNLOG**

The full path to *ndrxd* log file. Used by shell scripts.

**NDRX\_DMNLEV=LOG\_LEVEL\_OF\_NDRXD**

Log level of *ndrxd*. From 1..5. 5 is max. Used by shell scripts.

**NDRX\_LOG=FULL\_PATH\_TO\_XADMIN\_LOG\_FILE**

Full path to *xadmin* log file. Used by shell scripts.

**NDRX\_LEV=LOG\_LEVEL\_OF\_XADMIN**

Log level of *xadmin*. From 1..5. 5 is max. Used by shell scripts.

**NDRX\_XA\_RES\_ID=XA\_RESOURCE\_ID**

XA API two phase commit resource id. Must be set if using distributed transactions. Valid value is in range of 1..255.

**NDRX\_XA\_OPEN\_STR=XA\_OPEN\_STRING**

This is open string for two phase commit XA driver. Must be set if using distributed transactions.

**NDRX\_XA\_CLOSE\_STR=XA\_CLOSE\_STRING**

This is close string for two phase commit XA driver. This is optional, *NDRX\_XA\_OPEN\_STR* will be used if not set.

**NDRX\_XA\_DRIVERLIB=ENDUROX\_XA\_DRIVER\_LIB**

This is full path to Enduro/X shared library which loads the XA api.

**NDRX\_XA\_RMLIB=RESOURCE\_MANAGER\_SPECIFIC\_DRIVER**

This is full path to resource manager's specific driver (shared library). Must be set if using distributed transactions.

**NDRX\_XA\_LAZY\_INIT=LAZY\_INIT\_FLAG**

Set to 1 if XA sub-system should be initialized on first transactions. If not set, then defaulted to 0, meaning initialize XA sub-system on executable startup.

**NDRX\_XA\_FLAGS=XADMIN\_XA\_FLAGS**

Special for XA sub-system. It is semicolon separated tags with values. currently tag *RECON* is defined. *RECON* tag defines the number of attempts of *xa\_close()*/*xa\_open()* and doing *xa\_start()* again in case if original *xa\_start()* failed (the ATMI call *tpbegin()*) - for example firewall have been closed the connection. The format for the tag is: *RECON:<comma separated list of error codes e.g. 4,-8,\* - any err>:<number of attempts>:<sleep between attempts milli-sec>* example: *RECON:\*.3:100*, meaning reconnect on any *xa\_start* error, do the 3x attempts, sleep 100 milliseconds between attempts.

**NDRX\_NRSEMS=NDRX\_NRSEMS**

Number of semaphores used for protecting shared memory, when Enduro/X running in *poll()* mode. The minimum is recommended something about 7. Every service name in shared memory is hashed and semaphore protecting the service is calculated by modulus of NRSEMS. This affects *NDRX\_IPCKEY* semaphore, by giving the more occurrences in array. The first array entry is used by normal Enduro/X operations, and the others  $2..1+NRSEMS$  is used by *poll()* mode service protection. Default value is 30.

**NDRX\_MAXSVCSRVS=NDRX\_MAXSVCSRVS**

Max number of servers can advertise same service. This is used only in *poll()* and *SystemV* mode. The number affects the size of shared memory used for services. Bigger number causes more memory to be used for service registry. If the number of servers goes over this number for one service, the service entry will be ignored. Default is 30.

**PATH=PATH**

This is not Enduro/X specific env variable. But Enduro/X distribution bin directory should be included in system PATH env. variable.

**FLDTBLDIR=FULL\_OR\_RELATIVE\_PATH\_OF\_UBF\_FIELD\_DIRS**

This is colon separated list of directories where .fd files are located. I.e. UBF field definitions.

**FIELDTBLS=COMMA\_SEPERATED\_LIST\_OF\_FIELD\_FILES**

This is comma separated list of field files found in FLDTBLDIR.

**NDRX\_CCONFIG=NDRX\_COMMON\_CONFIG\_FILE**

If this is set then, all above configuration is read from specified ini file in *NDRX\_COMMON\_CONFIG\_FILE* in [*@global*] section. The Enduro/X config driving unit is able to merge configuration from multiple config files, and you can set higher priority files by in *NDRX\_CCONFIG1*, *NDRX\_CCONFIG2* *NDRX\_CCONFIG3*, *NDRX\_CCONFIG4*, *NDRX\_CCONFIG5*. Basically *NDRX\_CCONFIG* is lowest priority and *NDRX\_CCONFIG5* is highest priority. The specified configuration file can be directory, in that case Enduro/X will search for files with mask \*.ini, \*.cfg, \*.conf, \*.config files. All will be loaded in alphabetical order.

**NDRX\_CCTAG=NDRX\_COMMON\_CONFIG\_TAG**

This is basically subsection used for Enduro/X configuration sections [*@global*], [*@debug*], [*@queue*]. If the variable is set, then Enduro/X will lookup at process startup for sections like [*@global/YOUR\_TAG*], etc. CC tag can contain multiple sections, for example *server1/RM2*. Enduro/X will lookup the variables in each section [*@global/server1*] and [*@global/RM2*] for setting up the system.

**NDRX\_XADMIN\_CONFIG=XADMIN\_CONFIG\_FILE**

This variable is used by **xadmin** read the specific configuration file with xadmin's settings. Variable is optional.

**HOME=UNIX\_USER\_HOME\_DIR**

This variable is used by **xadmin** to search for per user configuration file when xadmin is started. In home directory search for configuration is done by *\$HOME/.xadmin.config*. This file is used in case if *NDRX\_XADMIN\_CONFIG* is missing. Variable is optional.

**VIEWDIR=NDRX\_VIEW\_DIR**

This is colon separated list of directories where to search VIEW compiled object files. The access to these directories are done only once operations with views are performed.

**VIEWFILES=NDRX\_VIEW\_FILES**

Comma separated list of VIEW object files (typically with extension .V). Object files are produced by view compiler **viewc(8)**.

**NDRX\_PLUGINS=NDRX\_PLUGINS**

This is semicolon separated string which denotes the list Enduro/X plugins which needs to be loaded at any XATMI program startup. Following plugins are provided with Enduro/X: libcryptohost.so - cryptography key by hostname.

**NDRX\_SILENT=SILENT\_SETTING**

If environment variable is present (and set to **Y**), the **xadmin** tool will not print banner header at startup.

**NDRX\_XADMINTOUT=XADMIN\_TOUT**

This override of **NDRX\_TOUT** configuration for xadmin queue operations. This timeout is used for certain calls to **ndrxd** (for example startup, shutdown, service listing, etc). Also it is used for communication with XATMI servers like **cpmsrv** and **tmsrv**. This is number of seconds, the value must be greater than 0. If parameter is set, then this will enable timeout control for communication with **ndrxd** daemon, if parameter is not set, the code will work in legacy mode, meaning that list calls to **ndrxd** will never get timeout.

**NDRX\_SVPROCNAME=SERVER\_PROC\_NAME**

Server process name exported by **ndrxd** at XATMI server process boot time. The name is either server name extracted by "<server>" or extracted by sub tag "<cmdline>".

**NDRX\_SVCLOPT=SERVER\_COMMAND\_LINE**

Server process command line. Generated and exported at the moment of XATMI server boot. If no command line options are passed to XATMI server, then **libatmisrv** tries to extract the parameter from this environment variable before failing, due to missing command line arguments.

**NDRX\_SVPPID=SERVER\_PARENT\_PID**

Parent process PID of server process. This process basically is the one which is booted by **ndrxd**. In case if server definition contains some wrapper processes (or scripts), then this basically is different than value of the real XATMI server. This variable is used by XATMI server library to report both PIDs to the **ndrxd**, the parent PID and the real process PID.

**NDRX\_SVSRVID=SERVER\_PROCESS\_ID**

This is XATMI server id set in <srvid> tag. The variable can be used for example is wrapper scripts to modify some resources used by process. For example if booting something like Tomcat app server, the admin TCP ports can be adjusted by this environment variable in order to avoid conflicts for booting multiple instances.

**NDRX\_SVQREADERSMAX=MAX\_SIMULTANEOUS\_READERS**

This configuration parameter set the maximum parallel readers for System V to Posix queues mapping tables. The number is used for read-write locks, thus the number sets the simultaneous readers, but during that time the write thread needs to wait for all readers to finish up the mapping when the writer will step in and all readers will wait. If the number is bigger read will be performed better when many processes are used, but that could lead to write starvation, and writes (opening queues) may become slow. System V queues acquires one more semaphore resource from the **NDRX\_IPCKEY**+1. Thus this semaphore is used read/write mode to protect the SystemV-to-Posix and Posix-to-SystemV mapping tables. The default value for this parameter is **50**.

**NDRX\_MSGQUEUESMAX=MAX\_IPC\_QUEUES**

Max number of queues that can be mapped to System V sub-system. This parameter defines size for two shared memory chunks which names are <**NDRX\_QPREFIX**>,shm,p2s and <**NDRX\_QPREFIX**>,shm,s2p. The number defines number of queue entries in the table. The bigger the number, the better hashing is got and lookup is quicker, but more memory is used. The minimum number shall be equal to the number of queues that will be used on the system, but recommended number is something as twice it. Each queue entry requires about 168 bytes. The default value for this parameter is **20000**.

**NDRX\_CLTTAG=CLIENT\_PROCESS\_TAG**

This is client process tag exported to **cpmsrv(8)** controlled processes. This value can be used also in **ndrxconfig.xml(5)** when formatting client process command line and log file names.

**NDRX\_CLTSUBSECT=CLIENT\_PROCESS\_TAG**

This is client process tag's sub-section exported to **cpmsrv(8)** controlled processes. This value can be used also in **ndrx-config.xml(5)** when formatting client process command line and log file names.

**NDRX\_LIBEXT=SHARED\_LIB\_EXTENSION**

This is shared library platform specific extension. For example Linux/AIX and Solaris would use "so". For MacOS it would be "dylib". Env variable is generated by provision script and it set in global variables.



**NDRX\_NORMWAITMAX=*NORM\_WAIT\_ATTEMPTS***

This configures number of attempts for **xadmin(8)** (and **tpadmsv(8)**) commands to wait in case if command requires normal **ndrxd(8)** context, but process is in other context, for example starting or stopping. The default is **60**. Between each attempt 1 second sleep is done.

## Chapter 3

# EXAMPLE

Sample configuration:

```
export FIELDTBLS=Exfields,fieldtab1.fd,fieldtab2.fd
export FLDTBLDIR=/enduro/tst1/tuxfbuf
export VIEWDIR=/enduro/tst1/views
export VIEWFILES=customer.V,card.V
export NDRX_DMNLEV=5
export NDRX_DMNLOG=/enduro/tst1/tmp/NDRXD
export NDRX_APPHOME=/enduro/tst1
export NDRX_CLUSTERISED=0
export NDRX_CMDWAIT=1
export NDRX_CONFIG=/enduro/tst1/conf/ndrxconfig.xml
export NDRX_DEBUG_CONF=/enduro/tst1/conf/ndrxdebug.conf
export NDRX_DPID=/enduro/tst1/tmp/ndrxd.pid
export NDRX_DQMAX=3000
export NDRX_HOME=/opt/endurox
export NDRX_IPCKEY=442000
export NDRX_LDBAL=0
export NDRX_LEV=5
export NDRX_LOG=/enduro/tst1/tmp/XADMIN
export NDRX_MSGMAX=100
export NDRX_MSGSIZEMAX=32000
export NDRX_NODEID=1
export NDRX_QPATH=/dev/mqueue
export NDRX_QPREFIX=/tst1
export NDRX_RNDK=jaUZwOlTqglSc
export NDRX_SHMPATH=/dev/shm
export NDRX_SRVMAX=10000
export NDRX_SVCMAX=600
export NDRX_TOUT=60
export NDRX_UBFMAXFLDS=16000
export NDRX_ULOG=/enduro/tst1/logs

# XA 2PC SECTION, ORACLE DB
export ORACLE_SID=ROCKY
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export PATH=$PATH:ORACLE_HOME/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export NDRX_XA_RES_ID=1
export NDRX_XA_OPEN_STR="ORACLE_XA+SqlNet=ROCKY+ACC=P/endurotest/endurotest1+SesTM=180+ ↵
    LogDir=/tmp/xa+nolocal=f+Threads=true"
export NDRX_XA_CLOSE_STR=$NDRX_XA_OPEN_STR

# XA Static registration driver
```

```
export NDRX_XA_DRIVERLIB=$NDRX_HOME/lib/libndrxxaoras.so

# XA Dynamic registration driver
#export NDRX_XA_DRIVERLIB=$NDRX_HOME/lib/libndrxxaorad.so

export NDRX_XA_RMLIB=/u01/app/oracle/product/11.2.0/dbhome_1/lib/libclntsh.so.11.1
export NDRX_XA_LAZY_INIT=1
# XA SECTION, END
```

Common-config ini file sample configuration, assuming following settings in env:

```
export NDRX_CCONFIG=/enduro/tst1/conf/app.ini
export NDRX_CCTAG=server1/RM2
```

The configuration file might look like:

```
#
# Common variables, inherited for CCTAGs
#
[global]
FIELDTBLS=Exfields,fieldtab1.fd,fieldtab2.fd
FLDTBLDIR=/enduro/tst1/tuxfbuf
VIEWDIR=/enduro/tst1/views
VIEWFILES=customer.V,card.V
NDRX_DMNLEV=5
NDRX_DMNLOG=/enduro/tst1/tmp/NDRXD
NDRX_APPHOME=/enduro/tst1
NDRX_CLUSTERISED=0
NDRX_CMDWAIT=1
NDRX_DPID=/enduro/tst1/tmp/ndrxd.pid
NDRX_DQMAX=3000
NDRX_HOME=/opt/endurox
NDRX_IPCKEY=442000
NDRX_LDBAL=0
NDRX_LEV=5
NDRX_LOG=/enduro/tst1/tmp/XADMIN
NDRX_MSGMAX=100
NDRX_MSGSIZEMAX=32000
NDRX_NODEID=1
NDRX_QPATH=/dev/mqueue
NDRX_QPREFIX=/tst1
NDRX_RNDK=jaUZwOlTqglSc
NDRX_SHMPATH=/dev/shm
NDRX_SRVMAX=10000
NDRX_SVCMAX=600
NDRX_TOUT=60
NDRX_UBFMAXFLDS=16000
NDRX_ULOG=/enduro/tst1/logs

#
# CCTAG section server1
#
[global/server1]
NDRX_CONFIG=/enduro/tst1/conf/ndrxconfig.xml

#
# CCTAG section RM2
#
[global/RM2]
# XA 2PC SECTION, ORACLE DB
ORACLE_SID=ROCKY
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
```

```
PATH=$PATH:ORACLE_HOME/bin
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
NDRX_XA_RES_ID=1
NDRX_XA_OPEN_STR="ORACLE_XA+SqlNet=ROCKY+ACC=P/endurotest/endurotest1+SesTM=180+LogDir=/tmp ↵
    /xa+nolocal=f+Threads=true"
NDRX_XA_CLOSE_STR=$NDRX_XA_OPEN_STR
# XA Static registration driver
NDRX_XA_DRIVERLIB=$NDRX_HOME/lib/libndrxxaoras.so
# XA Dynamic registration driver
#NDRX_XA_DRIVERLIB=$NDRX_HOME/lib/libndrxxaorad.so
NDRX_XA_RMLIB=/u01/app/oracle/product/11.2.0/dbhome_1/lib/libclntsh.so.11.1
NDRX_XA_LAZY_INIT=1
# XA SECTION, END

#
# Debug section (no need for NDRX_DEBUG_CONF in CConfig case)
#
[@debug]
*= ndrxx=0 ubf=0

[@debug/server1]
*= ndrxx=5 ubf=0
```

## Chapter 4

# NOTES FOR ORACLE DB

Note that **tmsrv** run with multiple threads. Flag *+Threads=true* MUST be set in **NDRX\_XA\_OPEN\_STR**. Otherwise unexpected core dumps can be received from **tmsrv**.

## Chapter 5

# BUGS

Report bugs to [support@mavimax.com](mailto:support@mavimax.com)

## Chapter 6

## SEE ALSO

`xadmin(8)`, `ndrxd(8)`, `ndrxconfig.xml(5)`, `ndrxdebug.conf(5)` `viewc(8)` `tpadmsv(8)`

## Chapter 7

# COPYING

© Mavimax, Ltd.