

# NDRXCONFIG.XML(5)

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>SYNOPSIS</b>	<b>1</b>
<b>2</b>	<b>DESCRIPTION</b>	<b>3</b>
<b>3</b>	<b>DEBUG CONFIGURATION FILE SYNTAX</b>	<b>4</b>
<b>4</b>	<b>EXAMPLE</b>	<b>8</b>
<b>5</b>	<b>BUGS</b>	<b>10</b>
<b>6</b>	<b>SEE ALSO</b>	<b>11</b>
<b>7</b>	<b>AUTHOR</b>	<b>12</b>
<b>8</b>	<b>COPYING</b>	<b>13</b>

## Chapter 1

# SYNOPSIS

```
<?xml version="1.0" ?>
<endurox>
  <appconfig>
    <sanity>SANITY_SECONDS</sanity>
    <checkpm>CHECKPM_STY</checkpm>
    <brrefresh>BRIDGE_REFRESH_TIME</brrefresh>
    <restart_min>MIN_RESTART_TIME</restart_min>
    <restart_step>RESTART_STEP</restart_step>
    <restart_max>MAX_RESTART_TIME</restart_max>
    <restart_to_check>NDRXD_RESTART_TO_CHECK</restart_to_check>
    <gather_pq_stats>NDRXD_GATHER_PQ_STATS</gather_pq_stats>
  </appconfig>
  <defaults>
    <min>MIN_SERVERS_DEFAULT</min>
    <max>MAX_SERVERS_DEFAULT</max>
    <autokill>AUTOKILL_DEFAULT</autokill>
    <env>ENV_OVERRIDE_DEFAULT</env>
    <start_max>MAX_STARTUP_TIME_DEFAULT</start_max>
    <pingtime>PING_EVERY_TIME_DEFAULT</pingtime>
    <ping_max>MAX_PING_TIME_DEFAULT</ping_max>
    <end_max>MAX_SERVER_SHUTDOWN_TIME_DEFAULT</end_max>
    <killtime>KILL_TIME_DEFAULT</killtime>
    <exportsvcs>EXPORT_SERVICES_DEFAULT</exportsvcs>
    <blacklistsvcs>BLACKLIST_SERVICES_DEFAULT</blacklistsvcs>
    <srvstartwait>NDRXD_SRV_START_WAIT_DEFAULT</srvstartwait>
    <srvstopwait>NDRXD_SRV_STOP_WAITT_DEFAULT</srvstopwait>
    <cctag>COMMON_CONFIG_TAG_DEFAULT</cctag>
    <protected>PROTECTED_SERVER_DEFAULT</protected>
    <reloadonchange>RELOAD_ON_CHANGE_DEFAULT</reloadonchange>
  </defaults>
  <servers>
    <server name="SERVER_BINARY_NAME">
      <min>MIN_SERVERS_SRV</min>
      <max>MAX_SERVERS_SRV</max>
      <autokill>AUTOKILL_SRV</autokill>
      <respawn>RESPAWN_SRV</respawn>
      <env>ENV_OVERRIDE_SRV</env>
      <start_max>MAX_STARTUP_TIME_SRV</start_max>
      <pingtime>PING_EVERY_TIME_SRV</pingtime>
      <ping_max>MAX_PING_TIME_SRV</ping_max>
      <end_max>MAX_SERVER_SHUTDOWN_TIME_SRV</end_max>
      <killtime>KILL_TIME_SRV</killtime>
      <sleep_after>SECONDS_TO_SLEEP_AFTER_SRV_START</sleep_after>
      <srvid>SERVER_ID</srvid>
    </server>
  </servers>
</endurox>
```

```

        <sysopt>ATMI_SERVER_SYSTEM_OPTIONS</sysopt>
        <appopt>ATMI_SERVER_APPLICATION_OPTIONS</appopt>
        <exportsvcs>ATMI_SERVER_EXPORT_SERVICES</exportsvcs>
        <blacklistsvcs>ATMI_BLACKLIST_SERVICES</blacklistsvcs>
        <srvstartwait>NDRXD_SRV_START_WAIT</srvstartwait>
        <srvstopwait>NDRXD_SRV_STOP_WAIT</srvstopwait>
        <cctag>COMMON_CONFIG_TAG</cctag>
        <protected>PROTECTED_SERVER</protected>
        <reloadonchange>RELOAD_ON_CHANGE_SERVER</reloadonchange>
    <fullpath>ATMI_SERVER_FULL_PATH</fullpath>
  </server>
  ...
  <server name="SERVER_BINARY_NAME_N">
    ...
  </server>
</servers>
<clients>
  <client cmdline="CLT_COMMAND_LINE [{${NDRX_CLTTAG} ${NDRX_CLTSUBSECT}}]"
    log="CLT_LOG"
    stdout="CLT_STDOUT"
    stderr="CLT_STDERR"
    env="CLT_ENV"
    CCTAG="CLT_CCTAG"
    wd="CLT_WD"
    klevel="CLT_KLEVEL"
    autostart="CLT_AUTOSTART">

    <exec tag="CLT_TAG_EXEC"
      subsect="CLT_SUBSECT_EXEC"
      log="CLT_LOG_EXEC"
      stdout="CLT_STDOUT_EXEC"
      stderr="CLT_STDERR_EXEC"
      env="CLT_ENV_EXEC"
      cctag="CLT_CCTAG_EXEC"
      wd="CLT_WD_EXEC"
      autostart="CLT_AUTOSTART_EXEC"
      klevel="CLT_KLEVEL_EXEC"
    />

    <exec tag="CLT_TAG_EXEC2"
      subsect="CLT_SUBSECT2_EXEC2"
      .../>

  </client>
  <client cmdline="BINARY2" ...>
    <exec tag="CLT_EXE_TAG2" .../>
  </client>
</clients>
</endurox>

```

## Chapter 2

# DESCRIPTION

*ndrxconfig.xml* holds the application domain configuration. It describes the ATMI servers which needs to be started. Counts of the, how much to start. Also it describes sanity times i.e. period after which system sanity checks should be made. Also it describes time frames in which ATMI server should start or stop. Internal server ping can be configured here too.

## Chapter 3

# DEBUG CONFIGURATION FILE SYNTAX

### **SANITY\_SECONDS**

Number of seconds after which perform system sanity checks. This number should divide by environment variable value *NDRX\_CMDWAIT*. As this actually is time by which *ndrxd* sleeps periodically.

### **CHECKPM\_STY**

This is number of sanity cycles into which check dead processes from the process model. This makes the actual checking of th PID existence system. Thus if *ndrxd* is started in learning mode and will not receive signals of the dead servers, then by setting it will discover exited processes.

### **BRIDGE\_REFRESH\_TIME**

Number of sanity units in which *tpbridge* refresh should be send to other node. If for example *SANITY\_SECONDS* is set to 10, and *BRIDGE\_REFRESH\_TIME* is set to 2 then period between bridge refreshes will be  $10 * 2 = 20$  seconds. Default value is **0** - do not send full updates.

### **MIN\_RESTART\_TIME**

Number of sanity units in which died server will be tried to start back. This is minimal time, means that this time is applied in case if server was running and died. If it is consecutive try, then *RESTART\_STEP* is applied on this timer.

### **RESTART\_STEP**

Number to sanity units to apply on *MIN\_RESTART\_TIME* in case of consecutive server death. Meaning that next try of restart will tried later that previous by this number of sanity units.

### **MAX\_RESTART\_TIME**

Max number of sanity units after which server will tried to restart. After each consecutive ATMI server death, next reboot is tried by  $MIN\_RESTART\_TIME + RESTART\_STEP * try\_count$ . If this goes over the *MAX\_RESTART\_TIME* then *MAX\_RESTART\_TIME* is used instead.

### **NDRXD\_RESTART\_TO\_CHECK**

Number of **seconds** for *ndrxd* to wait after daemon started in recovery mode. Within this time no sanity checks are performed, but instead "learning" mode is used. During this mode, *ndrxd* asks each ATMI server for it's configuration. If in this time ATMI server does not responds, then ATMI server is subject of sanity checks.

### **NDRXD\_GATHER\_PQ\_STATS**

Settings for **pq xadmin** command. if set to Y, *ndrxd* will automatically collect stats for service queues. In future this might be used for automatic service starting and stopping.

### **MIN\_SERVERS\_DEFAULT**

Default minimum number of copies of the server which needs to be started automatically. This can be overridden by *MIN\_SERVERS\_SRV* per server.

### **MAX\_SERVERS\_DEFAULT**

Max number of ATMI server copies per ATMI server entry. The difference between MIN and MAX servers means the number of standby servers configured. They can be started by hand with out system re-configuration. But they are not

booted automatically at system startup. You will have to start them with `$ xadmin start -s <server_name>` or by `$ xadmin start -i <server_id>`. This can be overridden by `MAX_SERVERS_SRV`.

#### **AUTOKILL\_DEFAULT**

Should server be automatically killed (by sequence signal sequence -2, -15, -9) in case if server have been starting up too long, or does not respond to pings too long, or it is performing shutdown too long. This can be overridden by `AUTOKILL_SRV` on per server basis.

#### **ENV\_OVERRIDE\_DEFAULT**

Full path to file containing environment variable overrides. see `ex_envover(5)` for more details. This can be overridden by per server basis by `ENV_OVERRIDE_SRV`. Both are optional settings.

#### **MAX\_STARTUP\_TIME\_DEFAULT**

Max time (in sanity units) in which server should start up, i.e. send init info to `ndrxd`. If during this time server have not initialized, it is being restarted. This can be overridden by `MAX_STARTUP_TIME_SRV`.

#### **PING\_EVERY\_TIME\_DEFAULT**

Number of sanity units in which perform peridical server pings. This can be overridden by `PING_EVERY_TIME_SRV`. Zero value disables ping.

#### **MAX\_PING\_TIME\_DEFAULT**

Number of sanity units, time in which server **must** respond to ping requests. If there is no response from server within this time, then restart sequence is initiated. This can be overridden by `MAX_PING_TIME_SRV`.

#### **MAX\_SERVER\_SHUTDOWN\_TIME\_DEFAULT**

Maximum time in which shutdown of server must complete in sanity units. If in given time server is not shutdown, then forced shutdown sequence is started until server exits. This can be overridden by `MAX_SERVER_SHUTDOWN_TIME_SRV` on per server basis.

#### **EXPORT\_SERVICES\_DEFAULT**

Comma separated list of services to be applied to all binaries which means the list of services to be exported by **tpbridge** server to other cluster node. This can be overridden by `ATMI_SERVER_EXPORT_SERVICES`.

#### **BLACKLIST\_SERVICES\_DEFAULT**

Comma separated list of services to be applied to all server binaries which means the list of services that must not be exported by **tpbridge** server to other cluster node. `ATMI_SERVER_BLACKLIST_SERVICES` is first priority over the `EXPORT_SERVICES_DEFAULT` if service appears in both lists. `BLACKLIST_SERVICES_DEFAULT` can be overridden by `ATMI_SERVER_BLACKLIST_SERVICES`.

#### **NDRXD\_SRV\_START\_WAIT\_DEFAULT**

Number of seconds to wait for servers to boot. If not started in given time, then continue with next server. This can be overridden by `NDRXD_SRV_START_WAIT`. Default value for this is 30 seconds.

#### **NDRXD\_SRV\_STOP\_WAIT\_DEFAULT**

Number of seconds to wait for server to shutdown. If not started in given time, then continue with next server. This can be overridden by `NDRXD_SRV_STOP_WAIT_DEFAULT`. Default value for this is 30 seconds.

#### **KILL\_TIME\_DEFAULT**

Time in sanity units after which to progress from first signal -2 to next signal -15. And after -15 this time means when next -9 signal will be sent. This is used if forced restart of forced shutdown was initiated by `ndrxd`. This can be overridden by `KILL_TIME_SRV`.

#### **COMMON\_CONFIG\_TAG\_DEFAULT**

Common configuration tag. Loaded into `NDRX_CCTAG` environment variable before process is spawned. This can be overridden by `COMMON_CONFIG_TAG`.

#### **PROTECTED\_SERVER\_DEFAULT**

Protected server is one that does not shutdown with `xadmin stop` unless you pass the `xadmin stop -c` paramter (complete shutdown). Still you can run the `sreload` and stop it by `xadmin stop -i <srvvid>` or by `xadmin stop -s <servernm>`. The `xadmin restart` won't work on these because `-c` is not supposed to be used by restart. The idea behind this, is to avoid accidental stop of the critical servers, like bridge or something else which is involved into `ndrxd` daemon management it self. This can be overridden by `PROTECTED_SERVER`.



**RELOAD\_ON\_CHANGE\_DEFAULT**

If set to **Y** or **y** the **ndrxd** daemon will scan the every binaries time stamp, and if it detects that time stamp is changed **ndrxd** will reload (stop/start) the XATMI servers one by one. The scanning will occur at every sanity cycle. This is recommended to be used **only** for development purposes. And must not be used on production servers! This can be overridden by **RELOAD\_ON\_CHANGE\_SERVER** on per server basis.

**SECONDS\_TO\_SLEEP\_AFTER\_SRV\_START**

Number of seconds to wait for next item to start after the server is launched. This is useful in cases when for example we start bridge server, let it for some seconds to connect to other node, then continue with other service startup.

**SERVER\_BINARY\_NAME**

ATMI server executable's name. The executable must be in \$PATH. This name cannot contain special symbols like path separator / and it cannot contains commas ,! Commas are used as internal queue separator combined with binary names.

**RESPAWN\_SRV**

Do the automatic process respawning if process is died for some reason. The default value is **Y**, meaning that processes are automatically recovered. If set to **N** or **n**, then sanity checks will not automatically re-boot the process.

**SERVER\_ID**

Server ID. It is internal ID for server instance. For each separate ATMI server the ID must be unique. Also special care should be take when **MAX\_SERVERS\_SRV** is greater than 1. In this case up till **MAX** servers, internally **SERVER\_ID** is incremented. Thus for example if **SERVER\_ID** is 200, and **MAX\_SERVERS\_SRV** is 5, then following server IDs will be reserved: 200, 201, 202, 203, 204. The maximum server id is set in **\$NDRX\_SRVMAX** environment variable. Minimal server id is 1.

**ATMI\_SERVER\_SYSTEM\_OPTIONS**

Command line system options passed to ATMI server. Following parameters are used by Enduro/X ATMI servers: **-N**, boolean type. If present, then no services will be advertised by server. In this case will be advertised only services specified by **-s** flag. For example if server advertises **SERVICE1**, **SERVICE2**, **SERVICE3**, but **-N** was specified, and **-sSERVICE3** is specified, then only service **SERVICE3** will be advertised. The **-s** argument also can contain aliases for services, for example **-sOTHERSVC:SERVICE2**, then new service **OTHERSVC** will be advertised which basically is the same **SERVICE2** (same function used). **-s** and **-N** can be mixed. **-s** can appear multiple times in system options. With one **-s** multiple services can be aliased to single existing service. The format is: **-s<NEWSVC1>/<NEWSVC2>/...<NEWSVCN>:<EXISTINGSVC>**. The **,** can be used as separator too, but for certain platforms it does not work, thus **/** is recommended. Server binaries output is controlled via **-e LOG\_FILE**, which means that stdout & stderr of server is dumped to **LOG\_FILE**. There are few internal params: param **-k** is just a random key for shell scripts. Another internal param is Server ID which is automatically passed to binary via **-i SERVER\_ID**. Enduro/X supports automatic buffer conversion for ATMI servers. Currently supported modes are **JSON2UBF**, **UBF2JSON**, these modes are activated by **-x** paramter in system options. These modes are passed for server functions being advertised. For example if we have service **functions** (not services) **UBF1FUNC**, **UBF2FUNC** and **JSONFUNC** and we want to ensure that these receive converted messages even if caller to **UBF** service sends **JSON** and vice versa, then following options might be set to command line: **-xUBF1FUNC,UBF2FUNC:JSON2UBF -xJSONFUNC:UBF2JSON**.

**ATMI\_SERVER\_APPLICATION\_OPTIONS**

Application specific command line options. This follows content after sys options as: *system options — app options*.

**ATMI\_SERVER\_EXPORT\_SERVICES**

Enduro/X server specific list of services to be exported. This list is only for **tpbridge** servers.

**ATMI\_SERVER\_BLACKLIST\_SERVICES**

Enduro/X server specific list of services that must not be exported. This list is only for **tpbridge** servers. Blacklist have higher priority over the Export list.

**ATMI\_SERVER\_FULL\_PATH**

This is full path of the XATMI server binary. At the porcess startup this overrides the server binary name at **SERVER\_BINARY\_NAME**. **ATMI\_SERVER\_FULL\_PATH** is used only for process startup. This is intended for testing, if server wrapper scripts needs to be started. But as the **ndrxd** will do the sanity checks against the process names, for time of the testing this needs to be disabled. Thus to do the testings with full path enabled, please increase the *checkpm* sanity unit time. As at the moment of process model checks, the **ndrxd** will find out that wrapped binary name does not contain the **SERVER\_BINARY\_NAME**, thus will reboot the process.

**CLT\_COMMAND\_LINE**

Executable name and arguments for client program. Command line basically is a format for subsection substitution. Other env variables available here too.

**CLT\_LOG**

Logfile to which stdout and stderr is logged. Can be overridden by *CLT\_LOG\_EXEC* for each individual process. Optional attribute.

**CLT\_STDOUT**

File where to log stdout. Can be overridden by *CLT\_STDOUT\_EXEC* for each individual process. Optional attribute.

**CLT\_STDERR**

File where to log stderr. Can be overridden by *CLT\_STDERR\_EXEC* for each individual process. Optional attribute.

**CLT\_ENV**

Environment override file. See **ex\_envover(5)** for syntax. Can be overridden by *CLT\_ENV\_EXEC* for each individual process. Optional attribute.

**CLT\_CCTAG**

ATMI Client lib Common-Config tag. Can be overridden by *CLT\_CCTAG\_EXEC* for each individual process. Optional attribute.

**CLT\_WD**

Working directory for the process. Can be overridden by *CLT\_WD\_EXEC*.

**CLT\_AUTOSTART**

Should process be started automatically? *Y* or *y* means boot at start. Can be overridden by *CLT\_AUTOSTART\_EXEC* for each individual process. Optional attribute. Default *n*.

**CLT\_TAG\_EXEC**

Tagname to be set for process.

**CLT\_SUBSECT\_EXEC**

Subsection to be set for process. - used as default.

**CLT\_KLEVEL**

Kill level of the client. **0** - do not kill child processes recursively of the client, **1** - do kill child processes only when performing SIGKILL (-9), **2** - do kill on SIGTERM and SIGINT child processes. The default is **0**.

---

## Chapter 4

# EXAMPLE

Sample configuration:

```
<?xml version="1.0" ?>
<endurox>
  <appconfig>
    <sanity>10</sanity>
    <brrefresh>6</brrefresh>
    <restart_min>1</restart_min>
    <restart_step>1</restart_step>
    <restart_max>5</restart_max>
    <restart_to_check>20</restart_to_check>
  </appconfig>
  <defaults>
    <min>1</min>
    <max>2</max>
    <autokill>1</autokill>
    <start_max>2</start_max>
    <pingtime>1</pingtime>
    <ping_max>4</ping_max>
    <end_max>3</end_max>
    <killtime>1</killtime>
  </defaults>
  <servers>
    <server name="tpevsrv">
      <srvid>14</srvid>
      <min>1</min>
      <max>1</max>
      <cctag>RM1</cctag>
      <env>${NDRX_HOME}/tpevsrv_env</env>
      <sysopt>-e /tmp/TPEVSRV -r</sysopt>
    </server>
    <server name="tpbridge">
      <max>1</max>
      <srvid>100</srvid>
      <sysopt>-e /tmp/BRIDGE -r</sysopt>
      <appopt>-n2 -r -i 0.0.0.0 -p 4433 -tA</appopt>
    </server>
    <server name="cpmsrv">
      <cctag>RM2</cctag>
      <srvid>9999</srvid>
      <sysopt>-e /tmp/cpmsrv.log -r -- -k3 -il</sysopt>
    </server>
  </servers>
  <clients>
```

```
<client cmdline="testbinary -t ${NDRX_CLTTAG} -s ${NDRX_CLTSUBSECT}" ↵
  autostart="Y" cctag="RM4">
  <exec tag="TAG1" subsect="SUBSECTION1" log="${APP_LOG}/testbin1-1. ↵
    log"cctag="RM5"/>
  <exec tag="TAG2" subsect="SUBSECTION2" log="${APP_LOG}/testbin1-2. ↵
    log"/>
</client>
<client cmdline="testenv.sh" env="environment.override1" log="env1.log">
  <exec tag="TESTENV" autostart="Y"/>
</client>
</clients>
</endurox>
```

## Chapter 5

# BUGS

Report bugs to [madars.vitolins@gmail.com](mailto:madars.vitolins@gmail.com)

## Chapter 6

## SEE ALSO

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

## **Chapter 7**

# **AUTHOR**

Enduro/X is created by Madars Vitolins.

## **Chapter 8**

# **COPYING**

© Mavimax, Ltd