

08 September 2003

By Dominique Chabord, dominique.chabord-at-bluedjinn.com

for Shaman-X project, <http://www.shaman-x.org>

Shaman-X To-Do-list

1. Wish list for WDX V0.6

WDX-030823-1

not planned
nice-to-have

Create easy to install kits for RedHat and Debian and release WDX 1.0

Test and finalization.

It represents some work to build the right RPM or .DEB; anyway, and we can stand as now with a clean .tar kit, as long as WDX is not widely used. The reason is because WDX has no dependency on other packages. It is a task we should ask RPM and Debian specialists to manage for us, for all future versions.

WDX-030823-2

31 Aug 2003

Done V0.4

Change command keywords

change option keyword -nfs by -waitmount

change option keyword -wait by -waitreserve

WDX-030823-3

not planned

Done V0.4

Modify the path to wdx mission files and separate different missions in different folders

/var/shaman-x/wdx/mission_name/mission_name.cfg

/var/shaman-x/wdx/mission_name/mission_name.loc

/var/shaman-x/wdx/mission_name/mission_name.sup

/var/shaman-x/wdx/mission_name/mission_name.log

/var/shaman-x/wdx/mission_name/newmasterscript

/var/shaman-x/wdx/mission_name/nosubstitutescript

/var/shaman-x/wdx/mission_name/exceptionscript

/var/shaman-x/wdx/mission_name/eventscript

/var/shaman-x/wdx/mission_name/mission_name.hot

/var/shaman-x/wdx/mission_name/mission_name.txt

/var/shaman-x/wdx/mission_name/snapshot.hot

/var/shaman-x/wdx/mission_name/snapshot.txt

This satisfies the requirement to make every mission independantly managed. Names of different scripts no longer need to include the mission name. Snapshots no longer risk to be affected to a wrong mission. The file mission-dat which is created at wdx process start and deleted at wdx process kill stays in /var/shaman-x/wdx/

WDX-030823-4

not planned
nice to have

Snapshot names are no longer limited to 8 characters

This limitation existed because old NT versions could use FAT16 file system which implies such a limitation.

People who, for any reason, use such a limited file system will themselves refrain from using non-MSDOS file names. WDX will no longer care about it.

The filename would be restricted to 20 characters (the same restriction as for Resource names) because of shared coding.

WDX-030823-5

not planned
nice to have

Add two new scripts:

RESIGNMASTERSCRIPT should be triggered on the masterholder when it becomes slave,

symmetrically to NEWMASTERSCRIPT

DETECTSUBSTITUTESCRIPT: should be triggered on the masterholder when the first substitute is detected, symmetrically to NOSUBSTITUTESCRIPT

When actions are programmed in NEWMASTERSCRIPT and NOSUBSTITUTESCRIPT, reverse actions should be programmed in these new scripts.

NEWMASTERSCRIPT and NOSUBSTITUTESCRIPT instrument the cases of failures. The consequences of the commands: -nomaster, getmaster need to use EVENTSCRIPT to stop the service which is restarted on the new masterholder (or substitute respectively).

This redefined set of four scripts make a consistent mechanism for basic implementation.

WDX-030823-6

not planned
nice to have

Add a list of optional parameters to exception command, passed as arguments to the exceptionscripts on all nodes

Parameters are strings, following a user defined policy, which are not controlled by WDX interface. WDX only limitates the length of the string, according to WDX design constraints.

Questions: What is this limit to fit in standard wdx message frame ? Is there a date associated to this action ?

It can be for example

- a predefined string in a user defined list of functions implemented in the exceptionscript:

ex: wdx mission -exception orderly-shutdown

- a predefined string and a value

ex: wdx mission -exception timerset 50

- a command suggested to all nodes (if string length is not limited)

ex: wdx mission -exception shutdown -r now

- a mix of all this and others

ex: wdx mission -exception Member_name wdx mission -reserve resource_name

in this last case exceptionscript will be triggered on all members:

mission.exceptionscript Member_name wdx mission -reserve resource_name

and the member which recognizes itself runs the command

This may be useful for example for DRBD management mission. DRBD is a network mirroring software. DRBD is based on a primary manager and a secondary manager. Only the primary manager knows whether the secondary disk copy is up to date or not, but the corresponding resource must be reserved or faulted by the secondary manager which handles this secondary disk.

If the primary manager sets resource status, the status of the resource is lost (set to free) upon primary manager failure; and this is BAD.

With to-day version, the masterholder must set its own resource and the substitute must react but setting an equivalent resource on its own.

Will there be a 20 characters limitation too ?

question

WDX-030823-7

not planned
nice to have

Incognito mode (aka witness mode):

Incognito is an additional mode (added to Masterholder/substitute and Satellite) Incognito supersedes parameters of mission.cfg and other modes. The node does NOT even need to be listed in the configuration file. Member_name is superseded by incognito keyword.

A dedicated mission.loc is available, according to which wdx starts local scripts, but never sends a message on the networks (neither watchdog messages or locking messages, or others). All commands which require to send a message are forbidden. Accepted commands are: daemon, join, leave, wait, nfs, list, version, namemaster,status, log, show.

Incognito member state is listed in -list output on an additional line. The result of -status command is incognito. If modification WDX-030823-7 is done, -status incognito network1 should result in the status of local machine interface: example: off

Incognito is a keyword. A wdx member cannot be called incognito

network interfaces are given in -daemon command as follows:

wdx mission -daemon KEYWORD NETWORK1 <NETWORK2>

where

KEYWORD = "incognito"

NETWORK1 = interface name, for example eth0. NETWORK1 supersedes the description of mission.cfg if it exists.

NETWORK2 = optionnal interface name, for example eth1. NETWORK2 supersedes the

where

KEYWORD = "incognito"

NETWORK1 = interface name, for example eth0. NETWORK1 supersedes the description of mission.cfg if it exists.

NETWORK2 = optional interface name, for example eth1. NETWORK2 supersedes the description of mission.cfg if it exists.

why is it useful ?

for several cases:

let's consider a redundant nfs server and client stations.

We can put WDX on all of them. clients can reconfigure path to new server upon failover (example)

But:

- we are limited to 16 members total in the mission

- a client can be switched on and switched off regularly and there is no reason for activation scripts on all others in this case

- we cannot add a new client without reconfiguring the mission on the critical servers (same for removing a client)

- if the user has root access on his workstation, he can make nasty things happen with WDX commands

- in far future I think we should be able to build quickly a crisis management station out of any kind of hardware available at the right place (imagine the station for crisis management is in a burning room, we must requisition a workstation anywhere else and set up WDX, MDX and HDX on it. It is not time to reconfigure all WDX missions on all members.

In all these cases, Incognito allows unlimited number of non critical systems to start scripts locally or display info upon critical system transitions. You even may consider using this for watching a stand alone server

WDX-030823-8

not planned

nice to have

Status command provides directly a specific information.

To identify the query, the status command accepts an optional parameter. The result is the line as extracted from -list display which begins by the parameter:

ex: wdx mission -status member_name answers

"member_name fault unknown unknown"

ex: wdx mission -status resource_name answers

"resource_name reserved member_name date-and-time-stamp"

WDX-030823-9

not planned

nice to have

Dynamic change mode command

This command changes the mode of the local member, from substitute to satellite or from satellite to substitute. This command supersedes information of configuration files, up to the reinitialization of the process which re-reads configuration files. If the member is IDLE, the command puts the member in running state.

wdx mission -satellite modifies member_mode from MASTER/SUBSTITUTE to SATELLITE. If member is idle, it becomes RUNNING

wdx mission -substitute modifies member_mode from SATELLITE to MASTER/SUBSTITUTE. If member is idle, it becomes RUNNING

As a consequence, the masterholder may trigger NOSUBSTITUTESCRIPT and DETECTSUBSTITUTESCRIPT scripts.

Today, WDX does not differentiate between substitutes which are able to take over masterholder's applications and those which are not. A substitute may be unable to take over if some resources are failed (access to shared data, access to user network...) In this case, the solution would be that substitute does not take part to competition for mastertoken This can be achieved through a -leave command on the substitute, but in this case the node no longer participates to the mission and network diagnostics for this mission. It must get out of any interaction with the mission.

An alternative is to calculate complex conditions with MathDoctor-MDX and distribute roles in satellite only missions.

WDX-030823-10

not planned

nice to have

WDX -list output should be time-stamped

New format line:

in satellite only missions.

WDX-030823-10

not planned
nice to have

WDX -list output should be time-stamped

New format line:

Status of mission "mission_name" by member "member_name" at date_and_time
(note: why are there theses " in this line ?)

Rational: redirect this output to a file with time identification.

WDX-030823-11

*31 Aug 2003
done V0.4*

Merge WDX common code for Linux and NT.

WDX-030823-12

not planned
nice to have

Port WDX to WindowsXP, with updated install procedure

WDX-030823-13

not planned
nice to have

Verification Test Procedures update and rebuild acceptance test scripts and documentation

Needs to be done if many changes are being made to the functionality of wdx and also for porting to NT and XP

WDX-030823-14

not planned
nice to have

Extension of wdx message.

The size of wdx message is limited to fit a single ethernet frame (1024 bytes total). It must be checked that nowadays the actual limit is 1500 bytes, and that wdx message length can be increased without any split on the network. This would allow to increase the number of resources in a mission or increase the length available for each resource.

WDX-030823-15

not planned
nice to have

Associate a value to a resource.

A dynamic value is associated to a resource and can be modified by the member <which has reserved it ?>.

the command is modified with a new keyword command -value as follows

wdx mission -value resource_name value

This causes all members to trigger eventscript with a new argument list. the output of -list is to be modified too.

This could be useful to manage timers through wdx, to characterize persistence, stability, delayed values and so on.

WDX-030823-16

not planned
nice to have

Improve mission.sup file functionality

Possibility to set several thresholds for a member.

WDX-080903-1

not planned
nice-to-have

Potential bug: The "User defined signal " message that is sometimes displayed

WDX-080903-2

not planned
nice-to-have

The appearance of the *syn and *asy files

To be checked, but it doesn't sound odd. It appeared when I used an unstable version and used kind of kill -9 and so on, and they are probably normal behavior.

WDX-080903-3

not planned
nice-to-have

Improve error messages

Command line error messages are confusing. For instance if the mission name in the "-start" command is mis-typed then the message "wdx process not started" is displayed. Would be better to say something like "Invalid mission, process not started". If the "-start" command is entered for a mission that is already started then the message "bad process status" is displayed - would be better to say "mission already started".

nice-to-have Command line error messages are confusing. For instance if the mission name in the "-start" command is mis-typed then the message "wdx process not started" is displayed. Would be better to say something like "Invalid mission, process not started". If the "-start" command is entered for a mission that is already started then the message "bad process status" is displayed - would be better to say "mission already started". Messages must be written for the user, not for the programmer. They must make sense for the user. We should modify them over time and English messages should be written by an English man. Messages can be more verbose whenever necessary

WDX-080903-4

not planned

nice-to-have

Add a message ID to error messages

A new way to manage messages more easily: each message is :

<message-ID><message comment>

<message_comment> is self explaining, and can be adapted when we think it is necessary

<message_ID> is a number which stays always the same, it may give a more precise info for problem finding. It is a key to user's guide also and we no longer need to modify the user's guide everytime we change the comment part. (keeping up the user's guide is close to impossible, today)

WDX-080903-5

not planned

nice-to-have

Internationalization

Messages texts could be centralized in a separate messages .h or .c file, we could build international versions (french german...) just by using the right message file and the right man pages. Messages are indexed by their message_ID

WDX-080903-6

not planned

nice-to-have

wdx -waitreserve and wdx -waitmount to be timed-out

WDX-080903-7

not planned

nice-to-have

Revisit Super-user privilege

About "this command requires Super user privilege "

How can a standard user use commands which do not require this authorization (list, status, nameserver, show) ?

Are we sure WDX actually makes the difference ?

Should we place something in /bin so a standard user can get access to wdx ?

Today I get back : "command not found".

WDX-080903-8

not planned

nice-to-have

Port WDX to Darwin (Mac OS X)

Recompile and validate WDX on Darwin platform

2. Sand Box for FWDX

FWDX should provide minimal functionality over a standard RS232 X cable, either because it detects it, or by configuration.

Possibility to tune heartbeat period and detection delay.

Implement a threshold capability with log file notice (shorter detection time with no script triggering). Useful for tuning, diagnose and support.

API for an application supervisor. The supervisor would send a message periodically and if the timer expires on the other end, the supervisorscript is started on the remote computer

FWDX could be speeded up by shortening the message that is sent over the RS232 link and removing the limitations on the send and receive timeout values. I think we might have to consider a real-time implementation of Linux. It can be similar to VMS and Digital Unix realtime kernel options (IIRC), or true real time implementation RT-Linux since we don't share the com port via any protocols. see additional comment below

We need to understand how "real-time" standard Linux can be. Up to now we've made no testing on loaded systems. Kernel 2.6 is supposed to improve scheduling, 2.2 was supposed to be rotten, 2.4 would be in the middle.

3. Sand Box for HDX

WDXGUI becomes HealingDoctor-HDX V0.1

Adapt it. Separate configuration, management and information interfaces

Add FWDX management

Consider a graphic presentation based on technical forms for every element

Consider a dedicated WDX mission for management interface

4. Sand Box for DDX

Evaluate VMware, BOCHS and UML frameworks