

# Starting User Programs on LynxOS, Linux and HP-UX computers

---

8th February 1994, updated 28th November 2006

---

To start processes or transfer files to a LynxOS, Linux or HP-UX system we use the transfer.ref system - an SL/CO invention but never mind.

SL LynxOS systems, disk based Linux systems and HPs have their transfer.ref files in `/user/pca/hostname/transfer.ref`. They are edited with your favourite editor.

Please note that on HP-UX systems you can also ask your HP-UX administrator to start your program in `/usr/opt/etc/rc.applic` or the SLOPS/SLAPS administrator to start it in `/user/slops/etc/rc` or with `smd`. Similarly on disk based Linux systems if you already have an init.d script to start your program (and you do not need it to be monitored by `clie`) then use it rather than setting it up in the transfer.ref.

AB LynxOS and Linux Front Ends also use the transfer.ref system however their transfer.ref files are created with the [Controls Configuration Database Portal](#), then choose AB Controls Configuration Applications (Oracle FORMS using JInitiator or Sun Java Plugin). You need to be enabled to login by the AB/CO/DM team. Inside the Java Forms you chose "Hardware". I will not try to describe how the transfer.ref entries are setup here. Once you have done all this the transfer.ref file is created for example for the sps machine v1n394 by logging into ABCOPL1 then:

```
cd /acc/dsc/src/sps/v1n494
make transfer.ref
```

The transfer.ref will be created in `/acc/dsc/sps/v1n494/etc/transfer.ref`.

For reference the list of accelerators is:

```
ade
cps
ctf
lei
lhc
lin
ln3
mcr
psb
sps
tst
```

To get an idea of how this is setup for the SL LynxOS machine svlsr2 have a look at:

</user/pca/svlsr2/transfer.ref>

which is accessible from all LynxOS and most HP-UX systems as simply `~pca/svlsr2/transfer.ref`. At the beginning you will see some weird syntax:

```
## /usr/local/bin/instdrvr -i NBC_PPCdesc
```

which we use to start synchronous things which absolutely must be done first such as loading 1553 drivers. After that we start transferring files like `/etc/passwd` to the RAM disk, then creating directories on the RAM disk if we have not already done so and then starting the various servers for the machine.

Transfer.ref files for [HP-UX](#) and [Linux](#) are similar and are also kept in `/user/pca/hostname/transfer.ref`.

In general it is [Alastair Bland](#) who sets all this up although [Enzo Genuardi](#), [Vladimir Glafirov](#) and [Nicolas De Metz-Noblat](#) can also do this if necessary or if they are the expert for a particular type of application.

It is best to send what you want transferring or starting as electronic mail in a syntax as close as possible to the file just mentioned.

If you use the *slaps/slops* or *common build* environment in your development then the best place to put your files, programs and servers is in */user/pcrops*. In general the binaries on a LynxOS, Linux or HP-UX system are not copied to the local hard disk by *upfiles/wreboot* although this can be arranged if really necessary. In general log files can be put in */tmp*. On SL LynxOS systems they will be deleted at the next reboot if they are in the */tmp* directory itself whereas they will not be deleted if they are in a sub directory of */tmp* on a system with a real hard disk (SL PC LynxOS) or an AB LynxOS system where */tmp* is on NFS. Disk based Linux systems clean */tmp* after a week or so. On HP-UX systems the recommended location for log files is apparently */usr/tmp*. The ideal place to send log files is really */dev/null* however ...

Once all this is setup in theory a reboot will start it all. We then monitor using [cluc](#) (and viewed in the CCC and offices using [xcluc](#)) that all programs started as a *server* actually run. If not *cluc* gives an error and the host will be in red on the *xcluc* display. We also monitor that the process has not forked too many times, that it is not taking too much CPU time and that it is not using too much memory.

With the *transfer.ref* system there are 4 user utilities:

- **killn processname**  
kill a process by its name  
Usage `killn [ -signo | -signame ] namelist`
- **psn processname**  
process list by approximate name (instead of typing `ps -ef | grep processname`)
- **upfiles -f filename**  
update a file

To get help on the syntax type: **upfiles**

Revision: 1.2 Date: 95/08/03

```
upfiles -all          - updates files & servers
upfiles -e           - List error exit values
upfiles -v           - Silent mode
upfiles -h           - The current help message
upfiles -f [ namelist ] - updates files
upfiles -o [ namelist ] - copy files, no date checked
upfiles -x [ namelist ] - updates files & execute associated actions
```

- **wreboot -n processname**  
kill, update and restart process by name (warm Reboot)

To get help on the syntax type: **wreboot**

Revision: 1.2 Date: 95/08/03

```
wreboot -h           - This help message
wreboot -L           - Classified list of references (example output on svlsr2)
wreboot -N [namelist] - Kill/restart servers
wreboot -n [namelist] - Kill/update/restart servers
wreboot -kill        - Kill all servers
wreboot -restart     - Kill/restart all servers
wreboot -v           - Silent mode
wreboot -d           - Debug child mode
wreboot -all         - Start all servers
```

The programs *upfiles* and *wreboot* use the information found in the *transfer.ref* file. In general you must be the owner of the file or process in question to use *upfiles* or *wreboot* for it. The CCC operators and root and *pca* can restart anything however.

## More detailed information

Good examples of what is done are in /user/pca/cs-ccr-cmw1/transfer.ref for Kris Kostro's programs. E.g.:

```
#####
# HISTORY:
# -----
# AUTHOR      DATE          MODIFICATION
# abl        09.may.03      Common Middleware machine for Kris Kostro
# kk         19.may.03      Enabled start of CMW Equip gateway
# kk         11.jun.03      Enabled start of CMW Equip gateway for EA
# abl        27.aug.03      changed cmwEasrv to run as bioper eacr for Vito Baggiolini
# abl        27.aug.03      added /spsea symbolic link to eanorth
# kk         15.dec.03      Added start of CMW logger (CMWlogger)
# kk         23.feb.04      Added start of Equip to CMW gateway for PS equipment ( equipCmwGw )
# kk         20.apr.04      Renamed cmwEquipSrv and cmwEaSrv to eqpgw_all and eqpgw_ea to reflect
Corba names
# abl        29.apr.04      added EA mtrio datamodules for pln351, pln494, p20d17
# abl        30.apr.04      added EA anacq datamodules for pln351, pln494, p20d17
# abl        10.jun.04      added kdsbal_forever to take trace of crash and reboot
# abl        23.jul.04      removed kdsbal_forever
# abl        27.jul.04      add symbolic link in /bin to /usr/opt/bin/clic
# abl        27.jul.04      clic ignores java programs
# kk         21.sep.04      added start of TS Equip gateway (for DIP) eqpgw_TS
# kk         08.oct.04      added start of CMW equip gateways for eqpgw_magea_n and eqpgw_magea_w
# kk         08.oct.04      added start of timev_multi to multicast ea events to ea equip gateways
# abl        14.oct.04      fixed timev_multi port (was 1951 now 1953)
# abl        22.oct.04      survey sosca4 and send result to stsrvt Clogger
# kk         26.oct.04      added start of mugef CMW server with modified config file (UDP port for
timev)
# abl        27.oct.04      use symbolic link to java method in DipGw shell script
# kk         09.nov.04      added forever loop for eqpgw_all
# abl        07.dec.04      removed PLC data modules
# abl        19.jan.05      setup eclipse using super javadev (not here) for Han Verhagen
# abl        01.mar.05      upgrade to SLC304, no java development
# kk         01.mar.05      removed eqpgw_magea_w, not needed anymore
# abl        31.mar.05      removed survey_sosca4 as it was for short hostname Clogger
# braathen 05.april.05     Lemom cluster comment added
# braathen 26.april.05     HP Health Agent added
# kk         28.apr.05      added start of alarmMonALL - CMW Alarm Monitor
# braathen 02.sept.05     updated Lemon install
#####
#
# source_dir source_file dest_dir dest_file own grp mode type version comm_line
.
.
/user/pca/bin/java/cmw CMWlogger /user/pca/bin/java/cmw CMWlogger pca accsoft 0755 server - %./
CMWlogger CMWlogger.sh%
# CMW gateway servers for SL-Equip devices
/user/pca/bin/i686/cmw eqpgw_all /user/pca/bin/i686/cmw eqpgw_all pca accsoft 755 server - %
export MALLOC_CHECK_=2;ulimit -c unlimited;while :;do ./eqpgw_all -config all.cfg;sleep 10;done%
/user/pca/bin/i686/cmw/spsea eqpgw_ea /user/pca/bin/i686/cmw/spsea eqpgw_ea spsea eacr 0755
server - %export MALLOC_CHECK_=2;ulimit -c unlimited;while :;do ./eqpgw_ea -config ea.cfg;
sleep 10;done%
/user/pca/bin/i686/cmw/spsea timev_multi /user/pca/bin/i686/cmw/spsea timev_multi spsea eacr
0755 server - % ./timev_multi -p 1953%
/user/pca/bin/i686/cmw/spsea eqpgw_magea_n /user/pca/bin/i686/cmw/spsea eqpgw_magea_n spsea
eacr 0755 server - %while :;do ./eqpgw_magea_n -config magea_n.cfg ;sleep 10;done%
/user/pca/bin/i686/cmw eqpgw_ts /user/pca/bin/i686/cmw eqpgw_ts pca accsoft 0755 server - %
```

```

export MALLOC_CHECK_=2;ulimit -c unlimited;while :;do ./eqpgw_ts -config ts.cfg ;sleep 10;done%
/user/pcrops/production/MugefCMWserver mugefSrv /user/pcrops/production/MugefCMWserver mugefSrv
opera aop 0755 server - %while :;do ./mugefSrv -config /user/pca/bin/i686/cmw/mugef.cfg;sleep
10;done%
# Dip<->CMW gateway, in Java
/user/pca/bin/java/cmw/dip DipGw /user/pca/bin/java/cmw/dip DipGw pca accsoft 0755 server - %
mkdir -p /tmp/log;./DipGw
# CMW server of PS-style alarms, in Java
# /user/pca/bin/java/cmw gtpm_mts /user/pca/bin/java/cmw gtpm_mts pca accsoft 0755 server - %./
gtpm_mts %
/user/pca/bin/java/cmw/alarmMon alarmMonALL /user/pca/bin/java/cmw/alarmMon alarmMonALL pca
accsoft 0755 server - %./alarmMonALL %
#
/user/pca/bin/i686 mhm /usr/opt/bin mhm root accsoft 4755 server - %./mhm -N4 -D110000%
# Equip server for PS CMW devices, all accesses using SFTPRO cycle
/user/pca/bin/i686/cmw equipCmwGw /user/pca/bin/i686/cmw equipCmwGw pca accsoft 0755 server -
%./equipCmwGw 20000 PS%

```

The main aim of the these lines is:

- restart the program at power on
- allow the actual owner or named operator accounts to restart the the program with "wreboot -n programname"
- [cluc](#) looks at the list started by wreboot and complains if any are not running
- [xcluc](#) shows clic complaints in red. An operator can then simply restart the offending program with the Xcluc Repair button.
- Document why and who wanted the program - this allows someone not familiar with the program and the machine to know what was expected and fix the problem

The "while do" stuff is for unreliable programs. And is only a bodge - to do it properly requires writing a real management shell script - the problem is that "wreboot -n eqpgw\_ts" above executed by an operator will start the while loop twice and the program may then run twice which could be very bad.

Java, nodal and shell scripts are usually handled by making a symbolic link in /tmp with a meaningful name (e.g. TI8\_logging) to the real binary (/usr/java/jdk/bin/java, /usr/local/bin/nodal or /bin/bash) and then invoking that with "/tmp/TI8\_logging -jar xxx.jar myclass -parameters". That way "wreboot" and "cluc" can find the process easily in the "ps -ef" output. This is needed for the monitoring and for wreboot to cleanly restart a program.

Please try to keep the command name less than 15 characters - otherwise it is difficult to find in the ps output, especially on LynxOS.

Please try to keep the total command line (what is between the % ... %) below 255 characters.

Together with the manually created SL style transfer.ref files there are now around 1000 Linux, HP-UX and LynxOS machines managed like this.

## Syntax of the transfer.ref file

- **Blank lines** are ignored.
- **#** : comments start with a hash character
- **#%** : On LynxOS and Linux Front Ends hash followed by percent indicates a synchronous command to be executed. This is typically used to install drivers etc. or anything else that must be done in a specific order. Note that setting an environment variable or a *cd* to a directory will not be remebered from one command line to another. This command execution is actually the method used to invoke **upfiles** and **wreboot** at boot time.
- **#% end** : a special command on LynxOS and Linux Front Ends that stops the parsing of the rest of the transfer.ref for further commands. This saves a bit of time on a slow machine if the transfer.ref is long.
- The rest of the lines have the format:

```
source_dir source_file dest_dir dest_file own grp mode type version comm_line
```

**type** values are:

- o **-** : file transfer, **source\_dir/source\_file** is copied to **dest\_dir/dest\_file** (if they have different dates **only**), the owner, group group and permissions of the file are set from **own**, **grp** and **mode**. Note that **version** should be - and the **comm\_line** should be empty (%%). Here is an example triggered by "**upfiles -f clic.cfg**":

```
/user/pca/default/clic i686 /usr/opt/etc clic.cfg root accsoft 0644 - - %%
```

- o **slink** : a symbolic link is made using the equivalent of "**ln -s source\_dir/source\_file dest\_dir/dest\_file**". Again **version** should be - and **comm\_line** should be %% . Here is an example triggered by "**upfiles -f java1.4**":

```
/local/java j2sdk1.4.2_08 /opt java1.4 root accsoft 755 slink - %%
```

- o **dir** : performs a directory copy. This is very difficult to use and should be avoided.
- o **command** : any copies like for **file** above will be performed, a **cd** to **dest\_dir** is done then the **comm\_line** will be executed. If there are any semi colons then it will be executed as user **own** by **/bin/sh**. Note that if there are no ";" characters in the line then a shell will not be invoked, it will be exced as a pure binary which means you cannot use shell syntax for pipes etc.!

The **version** field specifies an amount of time for **wreboot** to wait for the command to complete. Normally the aim of **wreboot** is to start all **command** and **server** entries in parallel as quickly as possible.

Due to a bug, certainly present in older versions of **wreboot**, shell commands tend to exit after one command so an elaborate work around has been found using a backgrounded sleep for the wait time, then the series of commands, followed by a kill of the sleep if the series of commands finished before the wait time.

Here is an example, triggered by "**wreboot -n yum**":

```
/usr/bin yum /usr/bin yum root root 0755 command 20 %sleep 20& /bin/rpm -q perl-XML-Simple || (date;./yum -y install perl-XML-Simple) 2>&1 | /usr/bin/tee -a /root/yum.log; kill $!%
```

A negative **version** field is used to specify a LynxOS real time priority at which to run the **comm\_line**.

- o **server** : used for starting servers which will be monitored by **clic**. In the initial **wreboot -all** at boot time (do **NOT** type this yourself!) no kill of the running server with the name corresponding to **dest\_file** is performed however normally any running server is killed then the copy like for **file** above will be done, followed by executing the command in **comm\_line**. This is the most important functionality of the **upfiles/wreboot** system. Here is a very simple example which starts the clic server, triggered by "**wreboot -n clic**":

```
/acc/dsc/mcr/Linux/bin clic /usr/opt/bin clic root bin 0755 server - %./clic -s%
```

- o **shell** : very similar to **command** above except that the **comm\_line** is only executed if **source\_dir/source\_file** needs to be copied to **dest\_dir/dest\_file** because they have different dates. It is rarely used. Note that this is triggered by "**upfiles -x motd.dyn**" in the following example:

```
/user/pca/motd.dyn default /etc motd.dyn root root 0644 shell - %cat /etc/motd.local /etc/motd.dyn >/etc/motd%
```

Note how it always the fourth field (**dest\_file**) that is used as the parameter for commands like "**wreboot -n dest\_file**", "**upfiles -f dest\_file**". If you do not supply it then all actions of the type will be performed. In other words if you type "**wreboot -n**" then you will kill and restart every **server** on the machine and execute every **command**! DO NOT DO THIS.

Already my attempts at explaining all of this have probably confused you. I could describe every tiny little detail but I would need to read all the source code (/acc/dsc/src/sl/wreboot) to be able to know myself what is done. So I leave it to you the curious reader!

## Links

A full detailed description of the **transfer.ref** system has not been written. However a description of how it fits in to the original SL way of doing things is in [Overview of the SL Control System from Application to Hardware](#) under the heading **How SL PowerPCs boot**.

For some more information refer to Installation and Management of the SPS and LEP Control System Computers ([ps](#) or [pdf](#)) which was presented at the Berlin Conference in October 1993.

---

[Alastair Bland](#)