**The New MAXDAT Environment In UNIX**

Note: *Anything in italics needs to be replaced with a value.* **Everything in bold is literal**

**Making the NEW environment from Scratch**

If you’re replacing an existing environment, skip this section and go to “Flow Of The Environment”.

In the new system, we have removed the ETL directory to make it easier and more uniform with all the directories. In the “base directory” that the System Admin will have to create (usually it is “/u01/maximus/maxdat-*environment code* for example, /u01/maximus/maxdat-dev) – you will create a “State Directory” (or project directory) and four directories for the systems files. In the “State” directory, create the 3 directories “scripts”, “logs” and “config”. You can do this with one command:

**mkdir scripts logs config**

then go into the “scripts” directory and make the directory “cronfiles”.

**mkdir cronfiles**

This directory was in the “home” directory in the old system which made it confusing because developers couldn’t see them. So, we’ve moved them here to be with the other scripts.

**Flow Of The Environment**

Problems arose with having environment variables in the system files “.profile” and “.bash\_profile”. First, sometimes it was unclear to some developers when the system was using .profile vs .bash\_profile. Secondly, there was also a problem that in some environments - these files were not being read when they should. Thirdly, when we needed changes, the admins unfamiliar with our processes were having to try and make the changes and we could not see the changes they made to verify them. So, we have reduced what needs to be in the .profile/.bash\_profile down to one line per project that won’t change. This modification also allows us to share application installations on the same server using the same login user – like we do with ILEB/NYEC.

You can test this setting by exiting once you’ve added it to the .profile/.bash\_profile and signing back in. Then do:

**env | grep MD**

and it should show you the line in the profile (If you use the standard variable name – see below).

All the shell scripts can now be found at:

**svn://rcmxapp1d.maximus.com/maxdat/trunk/corp/scripts**

And this document will explain the changes. (it is also be stored in SVN in the same directory)

PLEASE NOTE: This is a NEW paradigm for setting up the environment. You have to use the OLD way or the NEW way – but you CANNOT MIX the two! If the system has the Old Shell files and you need a small change, edit the Old version. If you go to the New version, you have to replace ALL of the “.sh” files. Once you’re in the New system, if you add and Old format shell script, it will break. So, do not use any of these files until you are ready to switch over to the new system.

**Building The New System**

First, you can remove all the MAXDAT related stuff from the user profile for an existing build. If you’re using Korn Shell (ksh) like in Texas, that is the “.profile” file. If you’re using the Bourne Alias Shell (bash) everywhere else, that is the “.bash\_profile” file. After you’ve removed all the old MAXDAT references, add this one line (for a new project – just add this line):

**export $MD\_SETENV=/***absolute\_path\_to\_the\_script\_directory***/.set\_env**

Example: export $MD\_SETENV=/u01/maximus/maxdat-dev/CADIR/scripts/.set\_env

Next, copy all the new “.sh” files into the scripts directory:

corp\_purge\_logs.sh

corp\_run\_bmp.sh

corp\_run\_connect\_test.sh

run\_kjb.sh

run\_ktr.sh

set\_env.txt

Additionally, if you are setting up EMRS or Production Planning, you’ll also need:

corp\_run\_emrs.sh

corp\_run\_planning.sh

Next, the files need to be converted using dos2unix and renamed using the “state code” (like corp\_run\_bpm.sh to tx\_run\_bpm.sh)

TIMEOUT:

There is also a file in the same directory called: d2u. It’s a utility I wrote myself. Put it in the script directory, do dos2unix to that file and make it executable one time – then you can use it as a shortcut. Just do:

**./d2u** *filename*

It will do dos2unix and replace the DOS file with the Unix file and will make it executable all in one step… and it’s easier to type… it’s there is you want to use it… I’ve had problems using it in Texas because we don’t have access to the sed command. But I’ve used it in all UK instance and I’ve tested it in IL/NY Dev and had no problems.

If you have no dos2unix utility – look in the d2u file and run each line as a Unix command replacing the $1 with d2u. Then d2u will work

Lastly, make sure all of your environment variables are correct. If you don’t like the vi editor, you can do these edits prior to putting the set\_env.txt file into Unix. Or, if you’re using putty you can create a new .set\_env file and copy/paste the file into vi which will convert it for you automatically.

So, here is the starting environment file. Rename it from “set\_env.txt” to “.set\_env” and you also have to do a dos2unix conversion on it too (if you don’t paste it into Unix using putty). Then make the following changes. Note that ALL of your environment setting will be in this one file. Also note that it says “change the lines with a -> at the beginning”, but I’ve also highlighted the part you need to change here in red with examples. See notes below.

# .set\_env - **[SITE]**

# =======================================================================

# Do not edit these four SVN\_\* variable values. They are populated when

# you commit code to SVN and used later to identify deployed code.

# $URL: $

# $Revision: $

# $Date: $

# $Author: $

# =======================================================================

#

# Set all of the exported environment variables in this file

# All lines marked with a "->" need to be set for the local server (and the "->" removed)

# ----- MAXDAT VARIABLES ----

export PENTAHO\_JAVA\_HOME=’**/u01/app/appadmin/product/java/jdk1.6.0\_31’**

export ENV\_CODE=uat #dev,apt,uat,prd

export STCODE=IL #TX,IL,NY,etc

export MAXDAT\_KETTLE\_DIR='**/u01/app/appadmin/product/pentaho/data-integration**'

export MAXDAT\_ETL\_PATH="**/u01/maximus/**maxdat-$ENV\_CODE/$STCODE/scripts"

export MAXDAT\_ETL\_LOGS="/u01/maximus/maxdat-$ENV\_CODE/$STCODE/logs"

export KETTLE\_HOME="**/u01/maximus/**maxdat-$ENV\_CODE/$STCODE/config"

export KTR\_LOG\_LEVEL='Basic'

export KJB\_LOG\_LEVEL='Detailed'

export INIT\_OK="$MAXDAT\_ETL\_PATH/${STCODE}\_run\_check.txt"

export CHILD\_FAIL="/tmp/${STCODE}\_child\_task\_fail.txt"

#

PATH="$KETTLE\_HOME/.kettle/kettle.properties:.:**/u01/app/appadmin/product/pentaho**

**/data-integration/**:/usr/bin:$PATH"

export PATH

#

# ---- EMRS VARIABLES -----

export EINIT\_OK="$MAXDAT\_ETL\_PATH/${STCODE}\_emrs\_init\_check.txt"

export EMRS\_DATA\_DIR=$MAXDAT\_ETL\_PATH/EnrollmentDataEMRS

#

# ---- MAIL VARIABLES -----

#mail related variables

export EMAIL='**MAXDatSystem@maximus.com**'

export EMAIL\_MESSAGE="/tmp/${STCODE}\_run\_bpm-ERROR-LOG.txt"

export EMAIL\_SUBJECT="Errors With ${STCODE}\_run\_bpm.sh in $ENV\_CODE"

#

# ---- P-PLANNING VARIABLES ----

export PLANNING\_OK="$MAXDAT\_ETL\_PATH/${STCODE}\_run\_planning\_check.txt"

export PLANNING\_FAIL="$MAXDAT\_ETL\_PATH/${STCODE}\_child\_planning\_fail.txt"

#

# ---- OTHER VARIABLES ----

export LOG\_LIFE\_DAYS=**30** # Number of days to keep log files before deleting

Notes:

* State Code needs to match whatever is used in the directory structure.
* There may be other directories you need to add to the PATH variable, depending on your installation.
* The EMAIL server setting should be good for any US Maximus site.

That’s it – the effort was made to make ALL of the “.sh” files generic. You should only have to change the “.set\_env” file. Oh, except for “corp\_run\_bpm.sh” file – you need to uncomment the modules you want to call (or add one that’s not there, but I tried to include all the ones we have so far), otherwise it’ll just execute Run Initialization and stop.

For those trouble modules where we have created a separate call script, let me know and I’ll write it/help you/review yours/whatever… to create a separate script that will work within this system.

**Sharing Environments**

In places where the environment is shared (like in NYEC/ILEB) where the Unix login is the same, you have to make one additional change. You can add as many environment project lines to the profile as you wish, but they all need to be unique variable names (I’d recommend a two-letter moniker for each project).

Example: If I were doing this for NYEC / ILEB and possibly other projects, the bash profile might look like this:

export $NY\_SETENV=/u01/maximus/maxdat-dev/NY/scripts/.set\_env

export $IL\_SETENV=/u01/maximus/maxdat-dev/IL/scripts/.set\_env

export $OH\_SETENV=/u01/maximus/maxdat-dev/OHIO/scripts/.set\_env

export $MA\_SETENV=/u01/maximus/maxdat-dev/MASSEB/scripts/.set\_env

Then you have one simple edit in the following files:

Corp\_purge\_logs.sh

Corp\_run\_bpm.sh

Corp\_run\_connect.sh

Corp\_run\_emrs.sh (if you’re using this file)

Corp\_run\_planning.sh (if you’re using this one)

And (for the “NY” example) change the line at the top from “source $MD\_SETENV” to “”source $NY\_SETENV”. So, your project would “know about” the other project .set\_env files, but it will only call the one for your project, and all your environment setting will be in that file – no shared settings.