

NAME

mst_envs - configure MST environment

SYNOPSIS

mst_envs [options]

OPTIONS

-csh, -tcsh, -ksh, -sh

output commands compatible with the specified shell

--define|-D option=value

Define a subsystem specific option; see <L/Subsystem Options> for more information.

--env environment variable name

Only output the specified environment variable. This option may be repeated.

--envs

only output the environment variables

--path

only output the path

--prepend

add MST executables to the front of the user's path (default behavior)

--append

add MST executables to the end of the user's path

--export, --noexport

whether the MST variables will be exported; defaults to **--export**

--fullpath, --nofullpath

whether the output PATH will include the user's PATH; defaults to **--fullpath** (yes)

--ldlibrarypath, --noldlibrarypath

whether the output will include the an LD_LIBRARY_PATH at all; defaults to **--noldlibrarypath** (no)

--pfx pfx

prefix the MST variable names with the given string.

--sysarch, --nosysarch

whether to output the system architecture variables; defaults to **--sysarch** (yes)

--help

print a short help message and exit.

--usage

print detailed usage instructions and exit.

DESCRIPTION

mst_envs is normally used inside a shell script (most likely the user's shell startup script, like *.cshrc* or *.profile*) to add the required entities to the user's environment to access the MST programs and facilities. **mst_envs** is designed to be `eval'd` by the shell, e.g.

```
eval `mst_envs -csh`
```

normally this results in the setting of various environment variables, including the user's path. To have the MST variables treated as shell variables, use the **--noexport** flag.

the **--sysarch** flag causes variables containing architecture specific info to be output. these variables are used in constructing paths in the MST environment. by outputting these variables, the user may make use of them at a later point without having to run the **sysarch** script separately.

In general, play with the options; it'll be pretty obvious what they do. Just don't **eval** the output until you've got what you want.

Subsystem Options

The **--define** option is used to provide extra options for various subsystems:

- `perl_version`

The version of Perl to enable access to. It may take the following values:

`path`

Use the Perl in the current path.

`default`

Use the Perl pointed to by `@MST_OTS_ROOT@/pkgs/perl`.

`version`

Use the Perl found at `@MST_OTS_ROOT@/pkgs/perl-version`.

The `MSTENVS_WRAP_perl` environment variable is set if the requested Perl isn't the default one so that the `@MST_ROOT@/bin/perl` wrapper will use that specified.

- `lua_version`

The version of Lua to enable access to. It may take the following values:

`path`

Use the Lua in the current path.

`default`

Use the Lua found in `@MST_OTS_ROOT@/system/bin/lua`.

`version`

Use the Lua found at `@MST_OTS_ROOT@/pkgs/lua-version`.

SPECIAL CASES

The `LD_LIBRARY_PATH` variable is set up to behave a bit differently from the other environment variables. Since we have chosen to avoid setting `LD_LIBRARY_PATH` in a user's environment in the past, if it is not present in the user's environment **mst_envs** will not output an `LD_LIBRARY_PATH`. If it is present, **mst_envs** will treat it much like it does `PATH` and `MANPATH`. Expect **--append**, **--prepend**, and **--nofullpath** to work as they would for `PATH`.

AUTHOR

Diab Jerius (djerius@cfa.harvard.edu)

M Tibbetts (mtibbetts@cfa.harvard.edu)