MOOT Tool Set

MOOT configuration viewing

/usr/local/online/MOOT/bin/rdbBrowse.exe is useful. Ask Joanne or Jim how to use...

MOOT configuration management

Tools are found in `$ONLINE_ROOT/LICOS_ETC/tools/mootTools`.
There are certain environment settings are needed, see Moot Notes for more info.

1) buildMootConfig.py:
Takes as arguments a list of files under the LICOS_Config/apps directory. In the trivial case, one file on the command line, it reads the file, finds the various LATC, LCI and LPA sections, and constructs a configuration set using the information from those sections. It then submits that set to the MOOT system and reports the generated configuration key (numerical). For configurations with multiple steps, this is done multiple times.
The syntax is:

> *panetta>> ./buildMootConfig.py <file> <keyName>*

or:

> *./buildMootConfig.py <file1> <file2> ... <fileN>*

in which case, the keyNames will be created from the filenames by removing the "`.xml`" from the end of the filename.

2) invalidateMootKey.py
Takes a series of integers as an argument, and marks each key INVALID. If the key does not exist, a warning message is logged.
This is a thin wrapper around `MootUpdate.killConfig(key)`. Example:

> *panetta>> ./invalidateMootKey.py 1531 1532 1533 1534 1535*
> *Invalidating key 1531... done*
> *Invalidating key 1532... done*
> *Invalidating key 1533... done*
> *Invalidating key 1534... done*
> *Invalidating key 1535... done*

3) invalidateMootConfigName.py **writeme**
Takes a configuration name and markes all keys for that name INVALID. This code is not written as yet, and may not be needed.

Creating a full set of MOOT configs from a LICOS_Config release.

There is one more tool in `$ONLINE_ROOT/LICOS_ETC/tools/mootTools`, not mentioned above: buildLicosConfig.py.
This tool refers to the file LICOS_Config.txt, which defines which MOOT configs to build. Simply running this script will build all the MOOT configurations listed in the txt file against $ONLINE_ROOT/LICOS_Config.