

# Extending the TriBITS TPL System

**Brent Perschbacher**



Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,  
for the United States Department of Energy under contract DE-AC04-94AL85000.





# Current Capabilities

- Simple TPLs
  - Consist of a static set of library files and possibly header files.
- TPLs with dependencies on other TPLs
  - Support hasn't been uniformly implemented. Can be done by declaring the dependent library's files as part of the TPL's own libraries and headers.
- Special cases like MPI
  - “TPL” by name, but treated differently than most TPLs.



# Current Needs

- Simple TPLs

We will still need these and they still need to be easy to add.

- TPLs with dependencies on other TPLS

TPLs must find their dependencies without duplication of code.

- Compiler option support

Features like OpenMP or CXX11 support should be treated as a dependency for packages to allow proper enabling/disabling.

- Testing if a compiler exists (Fortran)

Some packages like ForTrilinos require specific compilers. When the necessary compiler isn't available, those packages need to be disabled cleanly. Currently we have a hack for ForTrilinos.

- TPL versions

Packages sometimes require certain (minimum) versions of TPLs.



# Solution?

We were already planning on revamping our TPL system to use the CMake `find_package` call. That revamp hasn't happened yet, but could be altered slightly to support these “new” requirements.

- Call all non-package dependencies a “TPL” or `TP<?>`.
- `Find_package` just calls `Find<package name>.cmake`

Very general, allowing us to test for any feature, as long as we satisfy the expected `find_package` interface.

- Keep simple things simple

Adding a simple TPL should require 6 lines of code or less.

- Interface extensions

These will need to be standardized in some fashion.