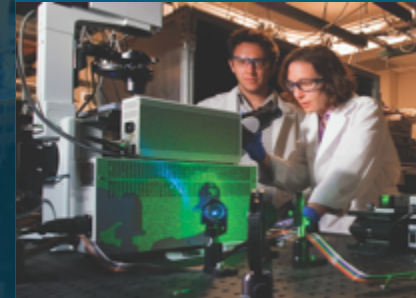


Componentized hierarchical build and test infrastructure and processes for CASL VERA



PRESENTED BY

Roscoe A. Bartlett, Ph.D.
Dept. Software Engineering & Research
Center for Computing Research
<https://bartlettroscoe.github.io>



The Challenge => Develop and Deploy Complex Software

- Multiple software repositories and distributed development teams
- Multiple compiled programming languages (C, C++, Fortran) and mixed-language programs
- Multiple development and deployment platforms (Linux, Windows, Super-Computers, etc.)
- Stringent software quality requirements

Solution Approach

=> TriBITS custom CMake build & test framework



Overview of CASL VERA Development

Overview of CASL



- **CASL: C**onsortium for the **A**dvanced **S**imulation of **L**ightwater reactors
- DOE Innovation Hub including DOE labs, universities, and industry partners
- Goals:
 - Advance modeling and simulation of lightwater nuclear reactors
 - Produce a set of simulation tools to model lightwater nuclear reactor cores to provide to the nuclear industry: **VERA: Virtual Environment for Reactor Applications**.
- Phase 1: July 2010 – June 2015
- Phase 2: July 2015 – June 2020
- Organization and management:
 - ORNL is the hub of the Hub
 - Milestone driven (6 month plan-of-records (PoRs))
 - Focus areas: **Physics Integration (PHI)**, Thermal Hydraulic Methods (THM), Radiation Transport Methods (RTM), Advanced Modeling Applications (AMA), Materials Performance and Optimization (MPO), Validation and Uncertainty Quantification (VUQ)

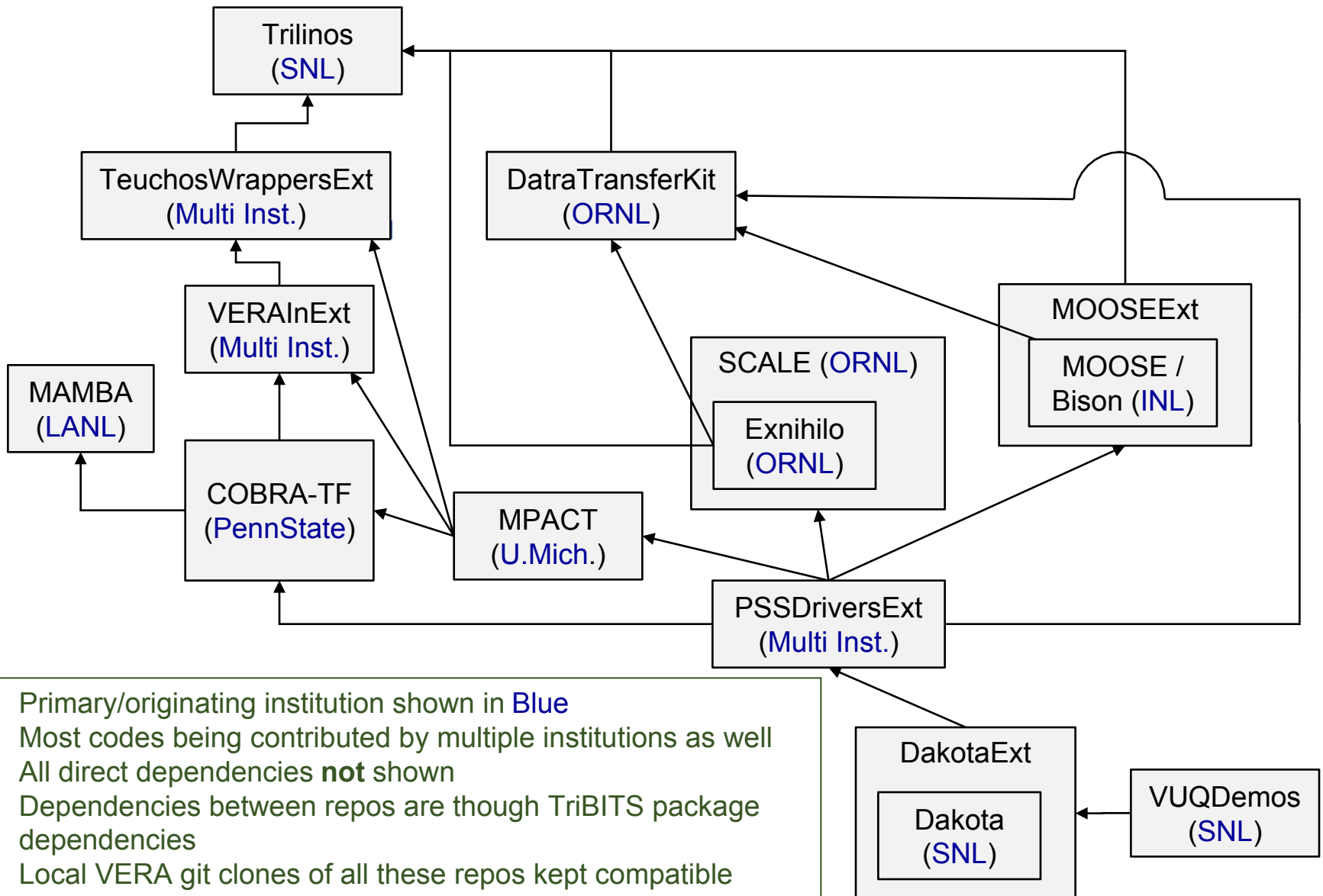
CASL VERA Development Overview (2015)



5

- VERA Development is complicated in almost every way ☹
- VERA Currently Composed of:
 - 18 different git repositories most with a different access list (NDAs, Export Control, IP, etc.)
- CMake build system using TriBITS Framework:
 - Over 2700 CMakeLists.txt files!
- VERA Software Development Process:
 - Official definition of VERA is 'master' branch of git repo mirrors at ORNL.
 - Primary development platform: ORNL CASL Machines (and clones)
 - VERA integration maintained by continuous and nightly testing:
 - Pre-push CI testing: checkin-test-vera.sh, cloned VERA git repos, on Fissile machine.
 - Post-push CI testing: CTest/CDash, all VERA git repos, shared libs.
 - Nightly CI testing: Debug and Release builds.
 - 100% passing builds and tests!
 - VERA snapshots and releases are taken off of 'master' branches on ORNL git repos.

Dependencies Between Selected VERA Repos (2015)





Why CMake?

Why TriBITS?



Open-source tools maintained and used by a large community and supported by a professional software development company (Kitware).

CMake:

- Simplified build system, easier maintenance
- Improved mechanism for extending capabilities (CMake language)
- Support for all major C, C++, and Fortran compilers.
- Automatic full dependency tracking (headers, src, mod, obj, libs, exec)
- Good Fortran support (parallel builds with modules with src => mod => object tracking, C/Fortran interoperability, etc.)
- Shared libraries on all platforms and compilers (support for RPATH)
- Faster configure times (e.g. > 10x faster than autotools)
- Native support for MS Windows (e.g. Visual Studio projects)
- Portable support for cross-compiling

CTest:

- Parallel running and scheduling of tests and test time-outs
- Memory testing (Valgrind)
- Line coverage testing (GCC LCOV)
- Better integration between the test system and the build system

Componentized CMake-based Projects Approaches



CMake, CTest, and CDash are great, **but raw usage does not scale very well to large projects and multiple repositories and teams!**

- **Multiple CMake projects:**
 - Manual builds and linking through <Package>Config.cmake files (e.g. Albany & Trilinos)
 - [CMake ExternalProject](#): (Also see [LLNL Spack](#))
 - + Provided as standard CMake module ExternalProject.cmake
 - + Allows non-CMake subprojects
 - Does not do full dependency tracking between component CMake projects
 - [Google Catkin](#) (used for the Google Robotics Operating System (ROS) project)
 - + Automatic dependency handling logic (implemented in Python)
 - + Parallel builds of CMake (sub)projects and posting to CDash
 - Requires Python for basic usage
 - [CMakeBasis](#) (Also see [LLNL BLT](#))
 - + Standardize CMakeList.txt and creation/usage of <Package>Config.cmake files
 - Core functionality requires Python
- **Single CMake project:**
 - Kitware [VTK Modules](#):
 - Does not support optional dependencies
 - Slow due to need to sort submodules based on dependencies
 - [TriBITS](#):
 - + Support multiple repos
 - + Core functionality depends only on CMake 3.10+

Why TriBITS?



- Framework for large, distributed multi-repository CMake projects
- Reduce boiler-plate CMake code and enforce consistency across large distributed projects
- Subproject dependencies and namespacing architecture (packages)
- Automatic package dependency handling (directed acyclic graph)
- Additional functionality missing in raw CMake
- Change default CMake behavior when necessary
- Additional tools for agile software development processes (e.g. Continuous Integration (CI))

History of TriBITS:

- 2007: Initially developed as a CMake package architecture for Trilinos
- 2011: Generalized and extended for CASL VERA
- 2014: Source code hosted on GitHub



Raw CMake vs. TriBITS

Example Raw CMakeLists.txt File



```
# Build and install library
set(HEADERS hello_world_lib.hpp)
set(SOURCES hello_world_lib.cpp)
add_library(hello_world_lib ${SOURCES})
install(TARGETS hello_world_lib DESTINATION lib)
install(FILES ${HEADERS} DESTINATION include)

# Build and install user executable
add_executable(hello_world hello_world_main.cpp)
target_link_libraries(hello_world hello_world_lib)
install(TARGETS hello_world DESTINATION bin)

# Test the executable
add_test(hello_world ${CMAKE_CURRENT_BINARY_DIR}/hello_world)
set_tests_properties(hello_world PROPERTIES PASS_REGULAR_EXPRESSION "Hello World")

# Build and run some unit tests
add_executable(unit_tests hello_world_unit_tests.cpp)
target_link_libraries(unit_tests hello_world_lib)
add_test(unit_test ${CMAKE_CURRENT_BINARY_DIR}/unit_tests)
set_tests_properties(unit_test PROPERTIES PASS_REGULAR_EXPRESSION "All unit tests passed")
```

**Executable and
test names must
be globally
unique!**

Example TriBITS Package CMakeList.txt File



```
tribits_package>HelloWorld)
TRIBITS_add_library(hello_world_lib HEADERS hello_world_lib.hpp SOURCES hello_world_lib.cpp)
TRIBITS_add_executable(hello_world NOEXEPREFIX SOURCES hello_world_main.cpp INSTALLABLE)
TRIBITS_add_test(hello_world NOEXEPREFIX PASS_REGULAR_EXPRESSION "Hello World")
tribits_add_executable_and_test(unit_tests SOURCES hello_world_unit_tests.cpp
    PASS_REGULAR_EXPRESSION "All unit tests passed")
tribits_package_postprocess()
```

- Less duplication and boiler-plate code
- Fewer commands
- **Build command wrappers:**
 - Install by default (most common)
 - Optionally Install libraries and headers or just executables?
 - Optional global prefixing of libraries
 - And more ...
- **CTest command wrappers:**
 - Automatic namespacing of tests and test executables
 - Classification of tests (BASIC, CONTINUOUS, NIGHTLY, ...)
 - Uniform handling of timeouts (and scaling of timeouts)
 - And more ...

Maintain consistency and add/change behavior across different independent repositories and packages and 1000s of CMakeLists.txt files!.



TriBITS Structural Units and Meta-Projects



TriBITS Project:

- Complete CMake “Project”
- Overall projects settings

TriBITS Repository:

- Collection of **Packages** and **TPLs**
- Unit of distribution and integration
- Typically a version control (git) repository

TriBITS Package:

- Encapsulated collection of related software & tests
- Unit of testing, namespacing, documentation, and reuse
- Lists dependencies on upstream **Packages** & **TPLs**

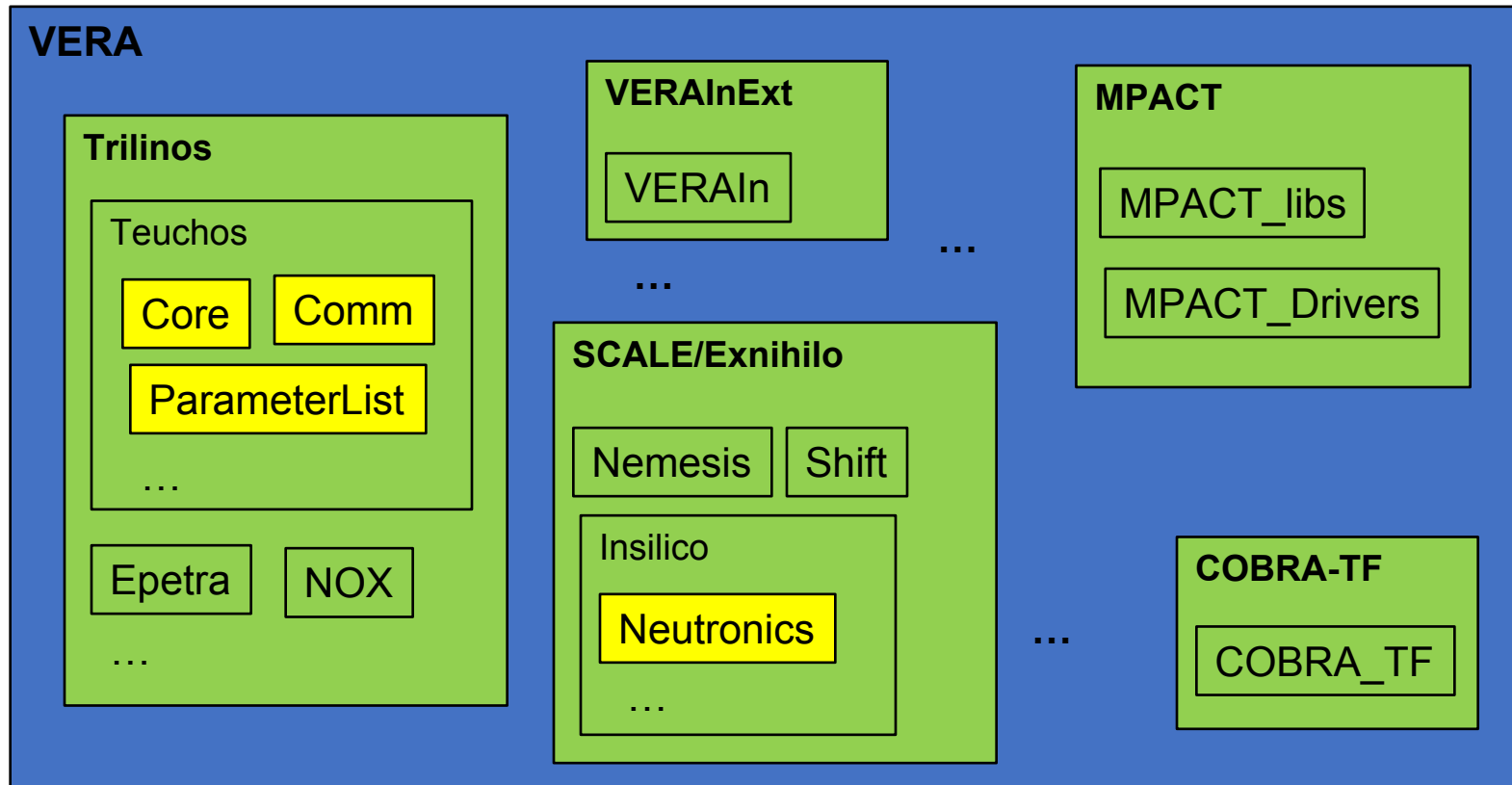
TriBITS Subpackage:

- Optional partitioning of package software & tests
- Primarily for dependency management (SE principles)

TriBITS TPLs (Third Party Libraries):

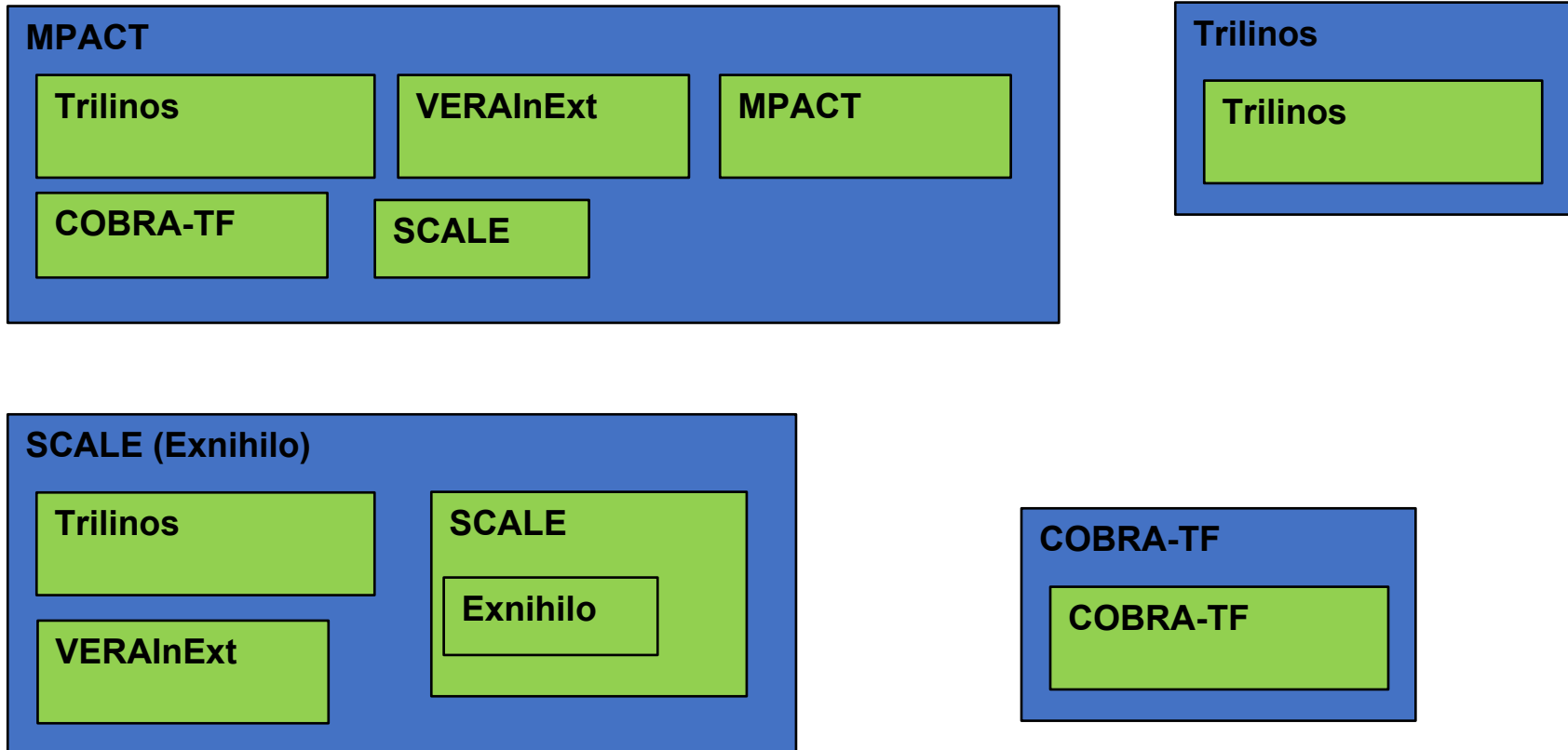
- Specification of external dependencies (libs)
- Required or optional dependency
- Single definition across all packages
- Can use standard CMake **find_package(<Package>)** and native **Find<Package>.cmake** modules

VERA Meta-Project, Repositories, Packages & Subpackages



- **VERA:** Git repository and TriBITS meta-project (contains no packages)
- **TriBITS and Git repos::** Trilinos, VERAInExt, COBRA-TF, MPACT, SCALE, Exnihilo ...
- **TriBITS packages:** Teuchos, Epetra, VERAIn, Insilico, COBRA_TF, MPACT_Drivers, ...
- **TriBITS subpackages:** TeuchosCore, InsilicoNeutronics ...
- **TriBITS SE packages:** Teuchos, TeuchosCore, VERAIn, Insilico, InsilicNeutronics, ...

Flexibility in TriBITS Projects and Repositories

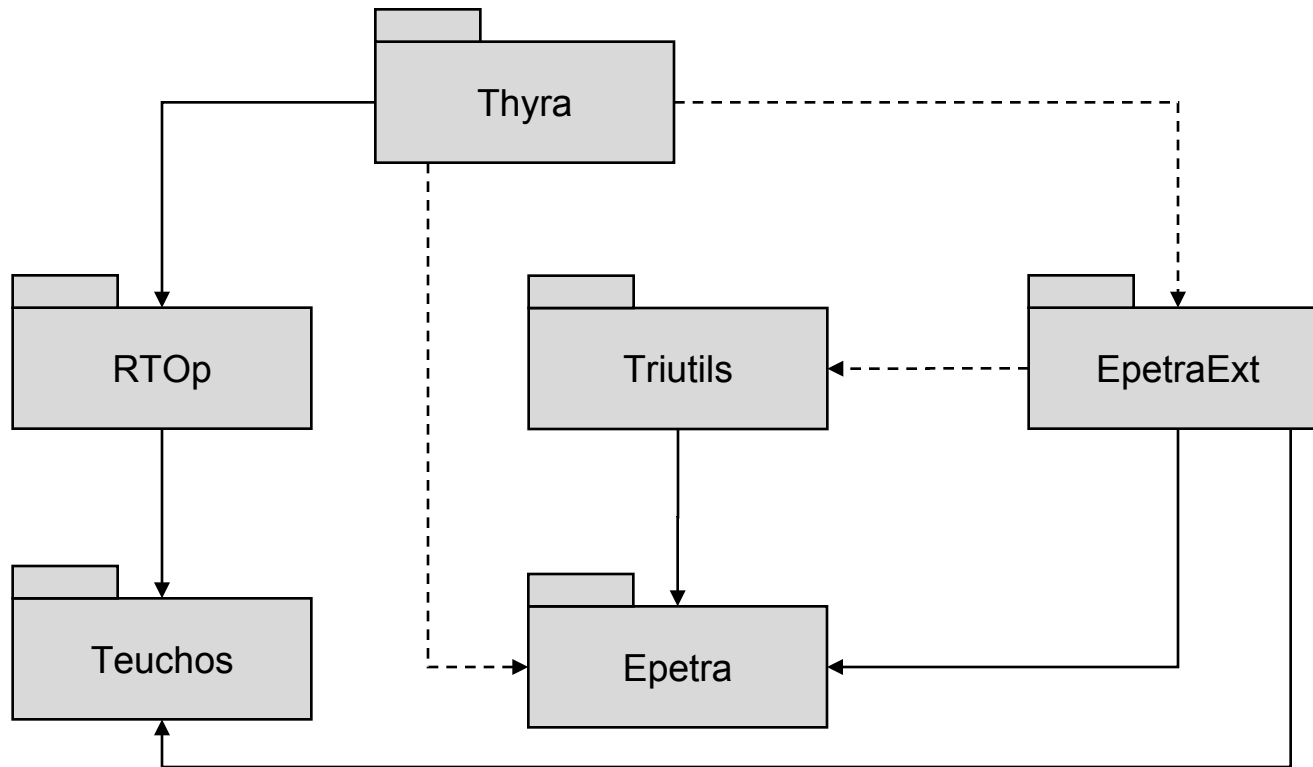


The same TriBITS repositories and packages can be arranged into multiple TriBITS projects.

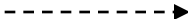


Automated Package Dependency Handling

Package Dependency Structure (e.g. Trilinos)



Required Dependence 

Optional Dependence 



Teuchos

```
tribits_package_define_dependencies(  
  LIB_REQUIRED_TPLS BLAS LAPACK  
  LIB_OPTIONAL_TPLS Boost )
```

Epetra

```
tribits_package_define_dependencies(  
  LIB_REQUIRED_TPLS BLAS LAPACK )
```

RTOp

```
tribits_package_define_dependencies(  
  LIB_REQUIRED_PACKAGES Teuchos )
```

Triutils

```
tribits_package_define_dependencies(  
  LIB_REQUIRED_PACKAGES Epetra )
```

EpetraExt

```
tribits_package_define_dependencies(  
  LIB_REQUIRED_PACKAGES Epetra Teuchos  
  LIB_OPTIONAL_PACKAGES Triutils )
```

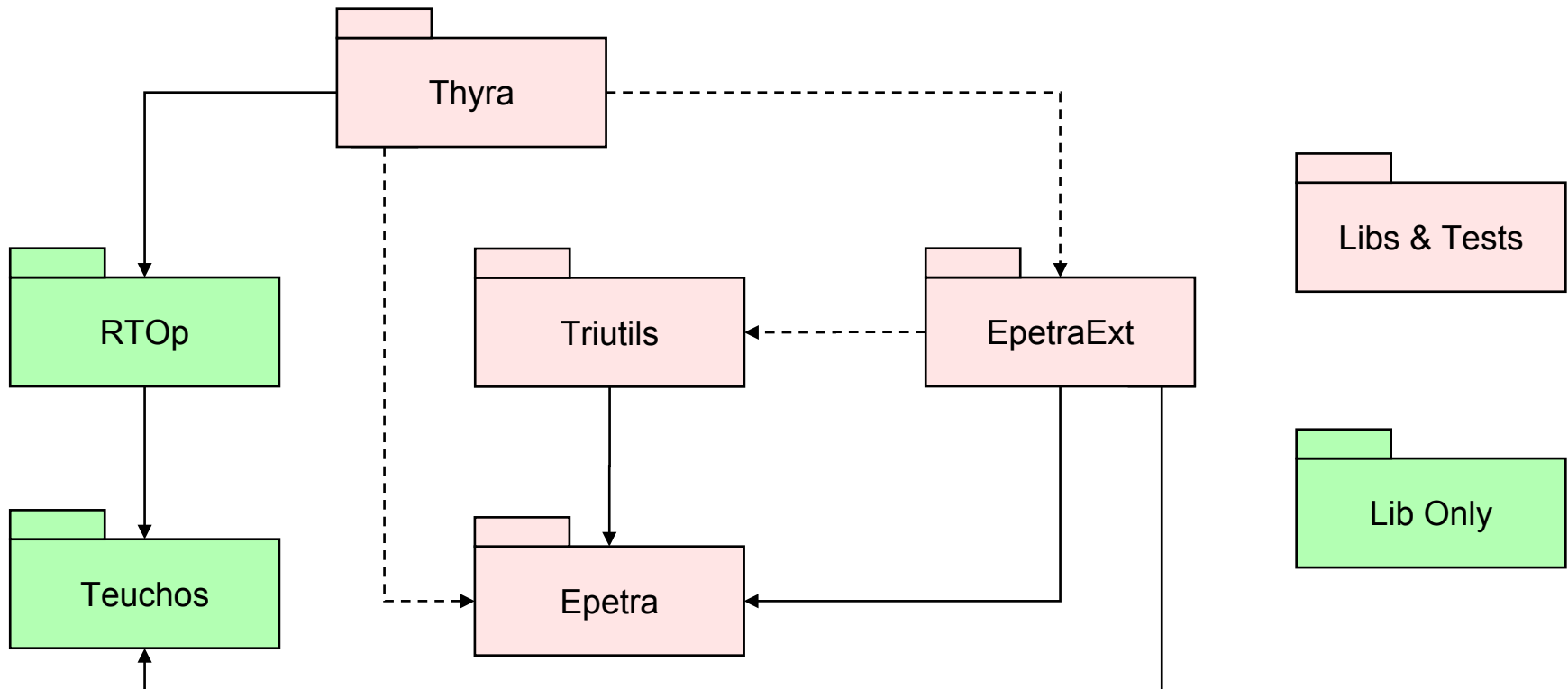
Thyra

```
tribits_package_define_dependencies(  
  LIB_REQUIRED_PACKAGES RTOp Teuchos  
  LIB_OPTIONAL_PACKAGES EpetraExt Epera )
```

CI Testing: Change Epetra



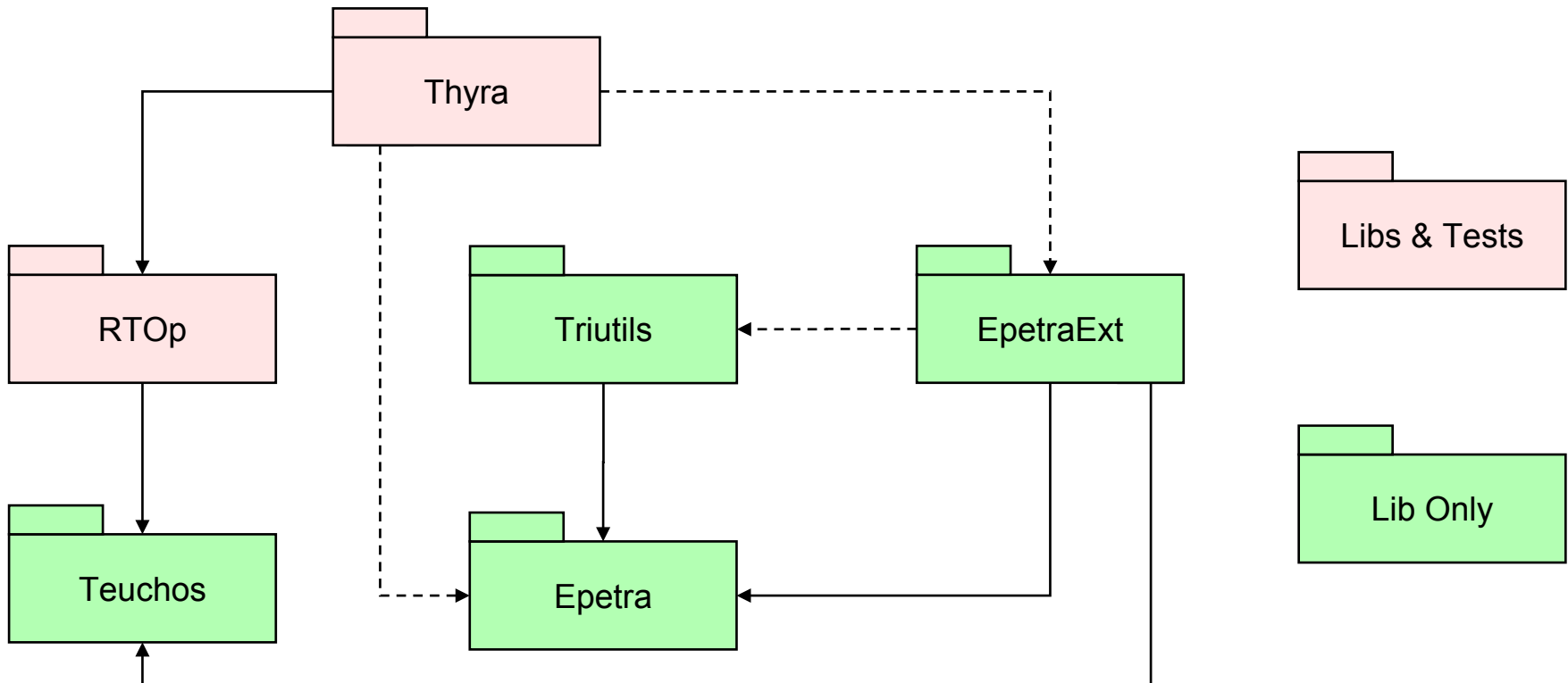
```
$ ./do-configure \  
-D Trilinos_ENABLE_Epetra=ON \  
-D Trilinos_ENABLE_ALL_FORWARD_DEP_PACKAGES=ON \  
-D Trilinos_ENABLE_TESTS=ON
```



CI Testing: Change RTOp



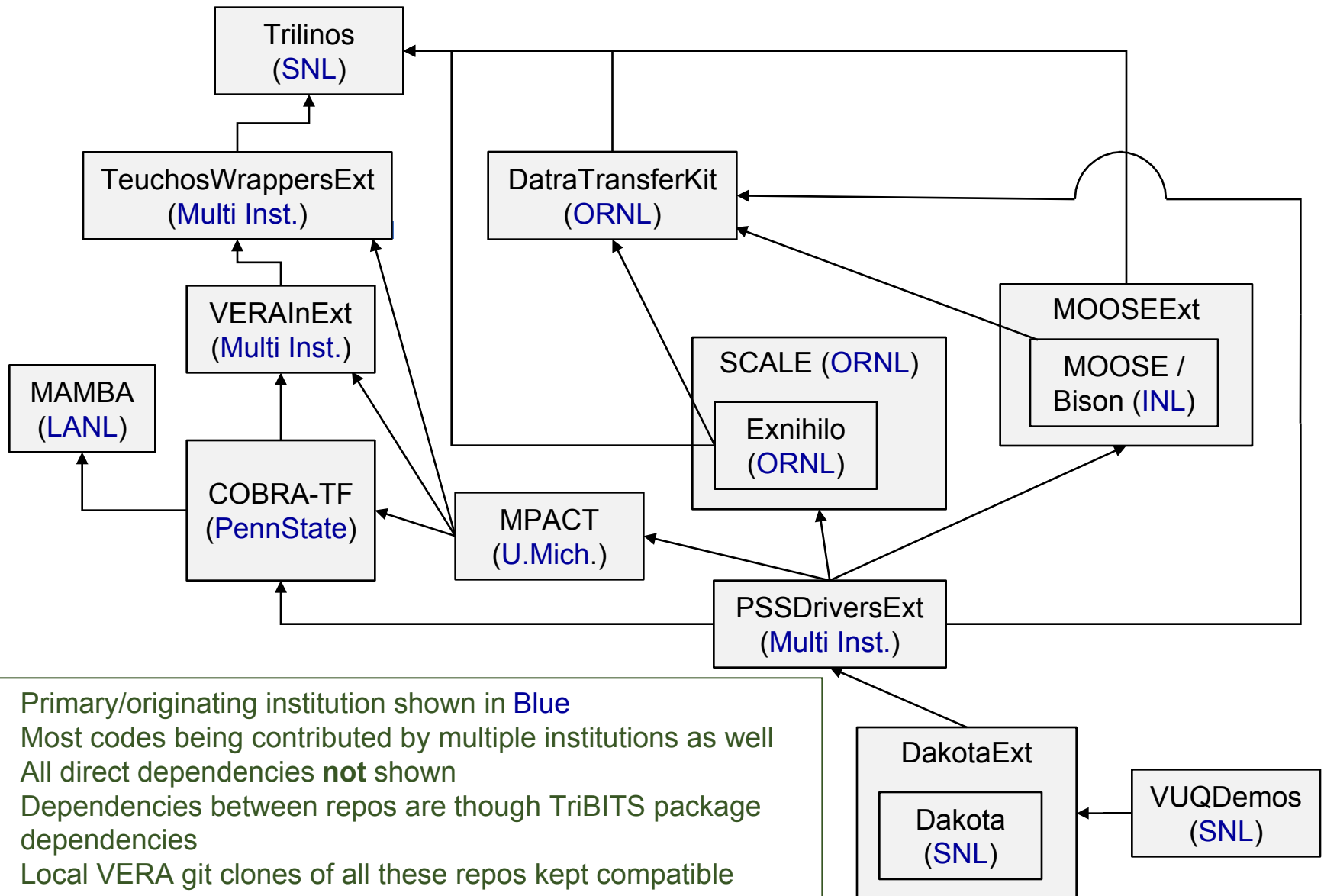
```
$ ./do-configure \  
-D Trilinos_ENABLE_RTOp=ON \  
-D Trilinos_ENABLE_ALL_FORWARD_DEP_PACKAGES=ON \  
-D Trilinos_ENABLE_TESTS=ON
```

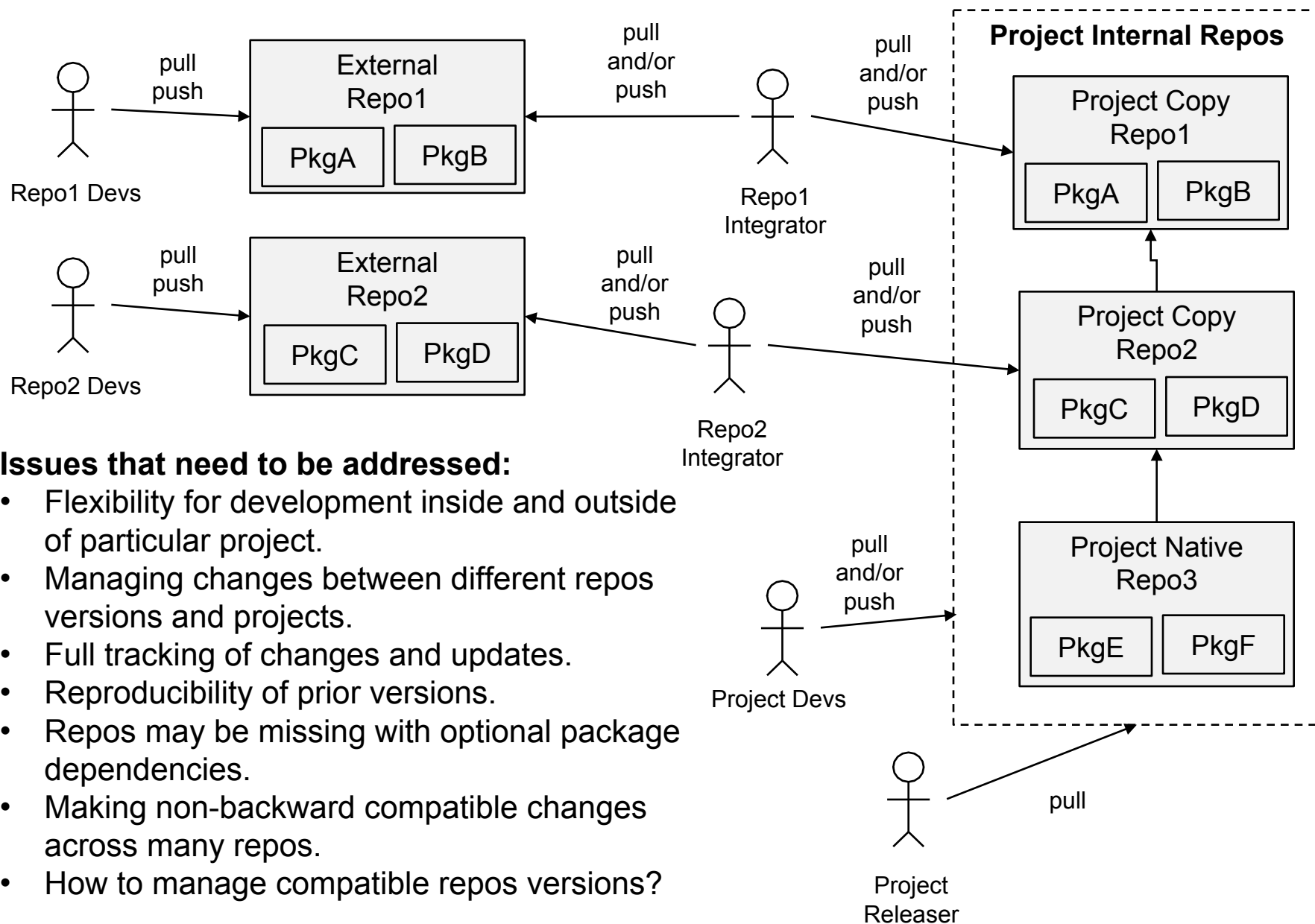




Multi-Repository Support

Dependencies Between Selected VERA Repos (2015)







```
tribits_project_define_extra_repositories(
  TriBITS          ""      GIT  git@casl-dev:TriBITS          ""      Continuous
  Trilinos          ""      GIT  git@casl-dev:Trilinos         ""      Continuous
  TeuchosWrappersExt ""    GIT  git@casl-dev:TeuchosWrappersExt ""    Continuous
  MAMBA             ""      GIT  git@casl-dev:MAMBA          ""      Continuous
  COBRA-TF          ""      GIT  git@casl-dev:COBRA-TF       ""      Continuous
  VERAINExt         ""      GIT  git@casl-dev:VERAINExt      ""      Continuous
  DataTransferKit   ""      GIT  git@casl-dev:DataTransferKit ""      Continuous
  MOOSEExt          ""      GIT  git@casl-dev:MOOSEExt       ""      Continuous
  MOOSE             MOOSEExt/MOOSE  GIT
    git@casl-dev:MOOSE  NOPACKAGES  Continuous
  SCALE             ""      GIT  git@casl-dev:SCALE         ""      Continuous
  Exnihilo          SCALE/Exnihilo  GIT
    git@casl-dev:Exnihilo          NOPACKAGES  Continuous
  MPACT             ""      GIT  git@casl-dev:MPACT          ""      Continuous
  LIMEEExt          ""      GIT  git@casl-dev:LIMEEExt       ""      Continuous
  hydrath           ""      GIT  git@casl-dev:hydrath        ""      Nightly
  PSSDriversExt     ""      GIT  git@casl-dev:PSSDriversExt  ""      Continuous
  DakotaExt         ""      GIT  git@casl-dev:DakotaExt      ""      Continuous
  Dakota  DakotaExt/Dakota  GIT  git@casl-dev:Dakota  NOPACKAGES  Continuous
  VUQDemos          ""      GIT  git@casl-dev:VUQDemos       ""      Nightly
)
```

Official version of VERA in on master branch used for CI & Nightly testing

- Partial set of repos can be cloned (protected by different groups)
- Non-git repos are converted into git repos: **Dakota**, **SCALE**, **MOOSE**



```
$ ./clone_extra_repos.py
```

```
...
```

ID	Repo Name	Repo Dir	VC	Repo URL	Category
1	TriBITS	TriBITS	GIT	git@casl-dev:TriBITS	Continuous
2	Trilinos	Trilinos	GIT	git@casl-dev:Trilinos	Continuous
3	TeuchosWrappersExt	TeuchosWrappersExt	GIT	git@casl-dev:TeuchosWrappersExt	Continuous
4	MAMBA	MAMBA	GIT	git@casl-dev:MAMBA	Continuous
5	COBRA-TF	COBRA-TF	GIT	git@casl-dev:COBRA-TF	Continuous
6	VERAIInExt	VERAIInExt	GIT	git@casl-dev:VERAIInExt	Continuous
7	DataTransferKit	DataTransferKit	GIT	git@casl-dev:DataTransferKit	Continuous
8	MOOSEExt	MOOSEExt	GIT	git@casl-dev:MOOSEExt	Continuous
9	MOOSE	MOOSEExt/MOOSE	GIT	git@casl-dev:MOOSE	Continuous
10	SCALE	SCALE	GIT	git@casl-dev:SCALE	Continuous
11	Exnihilo	SCALE/Exnihilo	GIT	git@casl-dev:Exnihilo	Continuous
12	MPACT	MPACT	GIT	git@casl-dev:MPACT	Continuous
13	LIMEExt	LIMEExt	GIT	git@casl-dev:LIMEExt	Continuous
14	hydrath	hydrath	GIT	git@casl-dev:hydrath	Nightly
15	PSSDriversExt	PSSDriversExt	GIT	git@casl-dev:PSSDriversExt	Continuous
16	DakotaExt	DakotaExt	GIT	git@casl-dev:DakotaExt	Continuous
17	Dakota	DakotaExt/Dakota	GIT	git@casl-dev:Dakota	Continuous
18	VUQDemos	VUQDemos	GIT	git@casl-dev:VUQDemos	Nightly

```
...
```

```
Running: git clone git@casl-dev:TriBITS TriBITS
```

```
Running: git clone git@casl-dev:Trilinos Trilinos
```

```
...
```

Only clones the repos that the user/developer has access to clone!



```
$ gitdist dist-repo-status
```

ID	Repo Dir	Branch	Tracking Branch	C	M	?
0	VERA (Base)	master	origin/master	2	1	
1	TriBITS	master	origin/master			
2	Trilinos	master	origin/master			
3	TeuchosWrappersExt	master	origin/master			2
4	MAMBA	master	origin/master			
5	COBRA-TF	master	origin/master			
6	VERAInExt	master	origin/master			3
7	DataTransferKit	master	origin/master			
8	MOOSEExt	master	origin/master			
9	MOOSEExt/MOOSE	master	origin/master			
10	SCALE	master	origin/master			
11	SCALE/Exnihilo	master	origin/master			
12	MPACT	master	origin/master	2		
13	LIMEExt	master	origin/master			
14	hydrath	master	origin/master			
15	PSSDriversExt	master	origin/master		4	
16	DakotaExt	master	origin/master			
17	DakotaExt/Dakota	master	origin/master			
18	VUQDemos	master	origin/master			

(tip: to see a legend, pass in --dist-legend.)

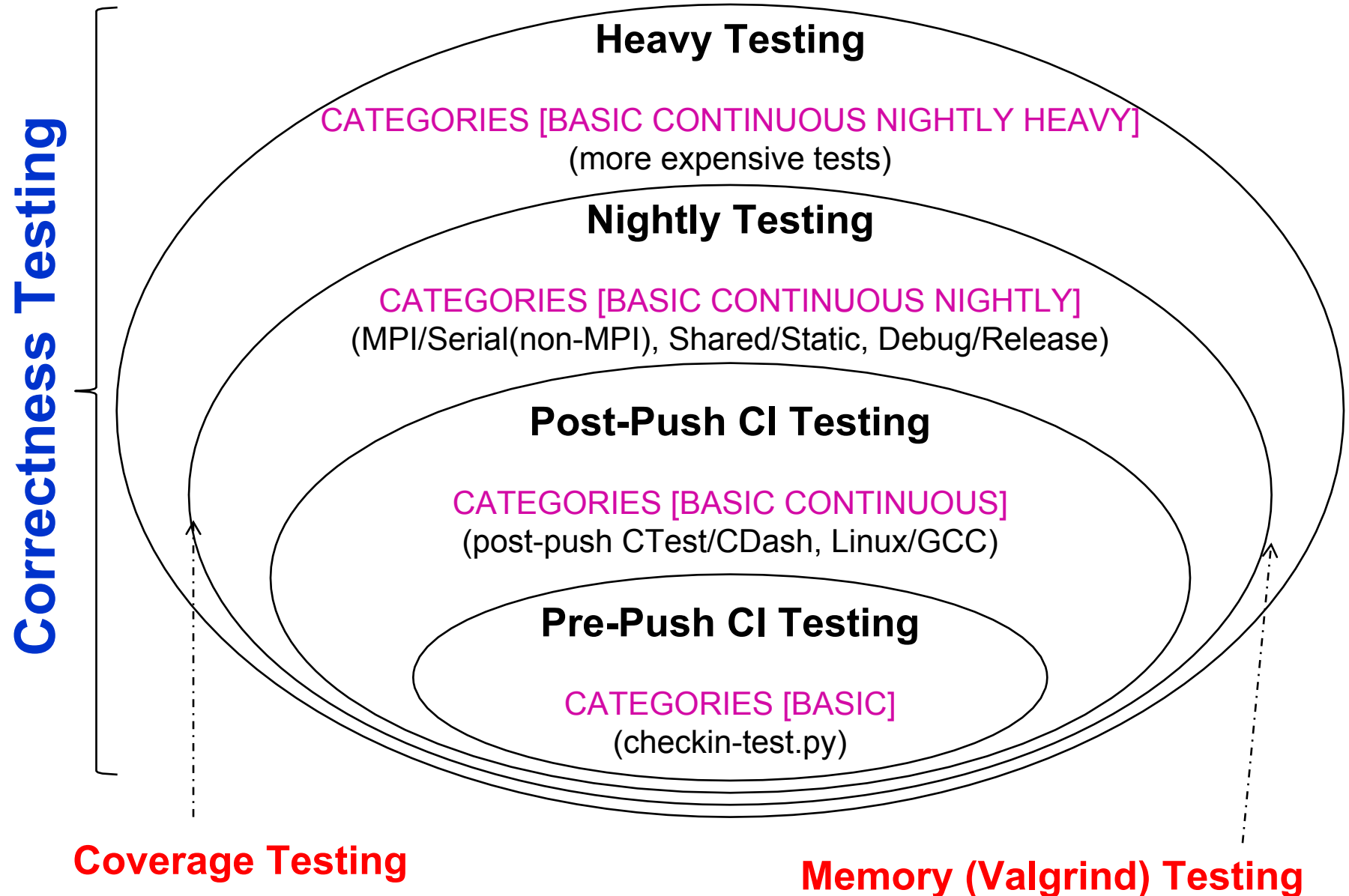


Testing Support

Standard TriBITS Test Categories and Layers



30





```
$ checkin-test.py --do-all --push
```

- Integrates with latest version in remote git repositories
- Figures out modified packages

Modified file: 'packages/teuchos/CMakeLists.txt'

=> Enabling 'Teuchos'!

- Enables all forward/downstream packages & tests
- Configures, builds, and runs tests
- Does the push (if all builds/tests pass)
- Sends notification emails
- Fully customizable (enabled packages, build cases, etc.)
- Documentation: `checkin-test.py --help`

CDash: VERA (Collapsed) Builds

32



[Login](#) [All Dashboards](#) Tuesday, June 07 2016 23:54:10

 **VERA**
Dashboard Calendar Previous Current Project

No file changed as of Tuesday, June 07 2016 - 00:59 EDT 

Filters [Help](#)

Match the following rule:

Site contains

Nightly

Site	Build Name	Update	Configure		Build		Test			Start Time ▼	Labels
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
	Linux-GCC-4.8.3-SERIAL_RELEASE_SHARED	0	0	72	0	224	0	0	1080	6 hours ago	(20 labels)
	Linux-GCC-4.8.3-SERIAL_RELEASE_DEBUG_SHARED	0	0	72	0	226	0	0	1080	10 hours ago	(20 labels)
	Linux-GCC-4.8.3-MPI_RELEASE_STATIC	0	0	65	0	251	0	1	1273	13 hours ago	(20 labels)
	Linux-GCC-4.8.3-MPI_RELEASE_DEBUG_STATIC	0	0	65	0	253	0	1	1273	15 hours ago	(20 labels)
	Linux-GCC-4.8.3-MPI_RELEASE_SHARED	0	0	65	0	252	0	0	1273	18 hours ago	(20 labels)
	Linux-GCC-4.8.3-MPI_RELEASE_SHARED_HEAVY	0	0	56	0	251	0	0	1796	22 hours ago	(19 labels)
	Linux-GCC-4.8.3-MPI_RELEASE_DEBUG_SHARED	0	0	65	0	254	0	0	1273	22 hours ago	(20 labels)
	Linux-GCC-4.8.3-MPI_DEBUG_DEBUG_SHARED	0	0	66	0	385	0	0	920	22 hours ago	(20 labels)

Continuous

Package-by-Package Build/Test Results

33

Site [REDACTED]
 Build Name: Linux-GCC-4.8.3-MPI_RELEASE_STATIC 
 Number of SubProjects Built: 20



Nightly													
SubProject	Update		Configure			Build			Test				Start Time 
	Files	Time	Error	Warn	Time	Error	Warn	Time	Not Run	Fail	Pass	Time	
VUQDemos	0	6s	0	9	14s	0	1 ⁺¹ ₋₁	42s	0	0	0	1s	5 hours ago
VUQCore	0	6s	0	9	16s	0	1 ⁺¹ ₋₁	48s	0	0	4	7s	5 hours ago
DakotaExt	0	6s	0	8	36s	0	1 ⁺¹ ₋₁	11m 6s	0	0	5	1m	5 hours ago
Tiamat	0	6s	0	3	4m 18s	0	13 ⁺⁷ ₋₇	2h 18m 12s	0	0	7 ⁺¹	27m 43s	7 hours ago
VeraAPI 	0	6s	0	3	41s	0	6 ⁺² ₋₂	54m 54s	0	1 ⁺¹	49	39m 42s	8 hours ago
MPACT_exe	0	6s	0	2	27s	0	0	27m 54s	0	0	180	23m 40s	9 hours ago
MPACT_Drivers	0	6s	0	2	13s	0	4 ⁺¹ ₋₁	11m 12s	0	0	5	9s	9 hours ago
MPACT_libs	0	6s	0	2	24s	0	15 ⁺¹ ₋₁	11m 24s	0	0	135	1m 55s	9 hours ago
XSTools	0	6s	0	2	6s	0	0	1m 18s	0	0	5	47s	9 hours ago
Insilico	0	6s	0	3	1m 2s	0	50	27m 54s	0	0	109	1m 42s	10 hours ago
CASL_MOOSE	0	6s	0	3	2m 55s	0	5 ⁺² ₋₂	1h 59m 12s	0	0	13 ⁺²	7m 46s	12 hours ago
DataTransferKit	0	6s	0	3	22s	0	27 ⁺¹ ₋₁	23m 54s	0	0	62	1m 59s	12 hours ago
VERAout	0	6s	0	2	5s	0	1 ⁺¹ ₋₁	54s	0	0	1	1s	12 hours ago
VERAIn	0	6s	0	2	9s	0	5 ⁺⁴ ₋₄	2m 36s	0	0	133	1m 26s	12 hours ago
COBRA_TF	0	6s	0	2	15s	0	91 ⁺⁴¹ ₋₄₁	27m 42s	0	0	344	26m 22s	13 hours ago
MAMBA	0	6s	0	2	5s	0	17 ⁺⁴ ₋₄	24s	0	0	4	0s	13 hours ago
Cicada	0	6s	0	2	8s	0	6 ⁺¹ ₋₁	2m 42s	0	0	5	21s	13 hours ago
ForTeuchos	0	6s	0	2	6s	0	7 ⁺⁴ ₋₄	1m 12s	0	0	4	1s	13 hours ago
CTeuchos	0	6s	0	2	9s	0	1	2m 48s	0	0	4	0s	13 hours ago
TriBITS	0	6s	0	2	28s	0	0	2m 36s	0	0	204	1m 58s	13 hours ago

CDash: VERA Package Builds (VeraAPI)



Login

All Dashboards

Tuesday, June 07 2016 23:51:19

CDASH

Dashboard

Calendar

Previous

Current

Project

VERA

No file changed as of Tuesday, June 07 2016 - 00:59 EDT

Nightly

Site	Build Name	Update	Configure		Build		Test			Start Time ▼	Labels
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
	Linux-GCC-4.8.3-SERIAL_RELEASE_SHARED	0	0	4	0	2	0	0	3**1	4 hours ago	VeraAPI
	Linux-GCC-4.8.3-SERIAL_RELEASE_DEBUG_SHARED	0	0	4	0	2	0	0	3**1	7 hours ago	VeraAPI
	Linux-GCC-4.8.3-MPI_RELEASE_STATIC	0	0	3	0	6_2	0	1**1	49	8 hours ago	VeraAPI
	Linux-GCC-4.8.3-MPI_RELEASE_DEBUG_STATIC	0	0	3	0	6_2	0	1**1	49	9 hours ago	VeraAPI
	Linux-GCC-4.8.3-MPI_RELEASE_SHARED_HEAVY	0	0	3	0	6_2	0	0	54	11 hours ago	VeraAPI
	Linux-GCC-4.8.3-MPI_RELEASE_SHARED	0	0	3	0	6_2	0	0	49	13 hours ago	VeraAPI
	Linux-GCC-4.8.3-MPI_RELEASE_DEBUG_SHARED	0	0	3	0	6_2	0	0	49	17 hours ago	VeraAPI
	Linux-GCC-4.8.3-MPI_DEBUG_DEBUG_SHARED	0	0	3	0	8_4	0	0	14	18 hours ago	VeraAPI

CDash: VERA Packages


[Login](#) [All Dashboards](#)

Tuesday, June 07 2016 23:42:21


[Dashboard](#)
[Calendar](#)
VERA
[Previous](#)
[Current](#)
[Project](#)

Project

Project	Configure			Build			Test			Last submission
	Error	Warning	Pass	Error	Warning	Pass	Not Run	Fail	Pass	
VERA	0	159	159	0	132	27	0	2	9968	2016-06-07 21:44:04

SubProjects

SubProject	Configure			Build			Test			Last submission
	Error	Warning	Pass	Error	Warning	Pass	Not Run	Fail	Pass	
CASL_MOOSE	0	8	8	0	6	2	0	0	73	2016-06-07 18:20:49
Cicada	0	8	8	0	6	2	0	0	30	2016-06-07 17:23:49
COBRA_TF	0	8	8	0	8	0	0	0	2666	2016-06-07 17:24:08
CTeuchos	0	8	8	0	8	0	0	0	28	2016-06-07 17:21:17
DakotaExt	0	8	8	0	8	0	0	0	41	2016-06-07 19:26:33
DataTransferKit	0	8	8	0	8	0	0	0	434	2016-06-07 17:58:42
ForTeuchos	0	8	8	0	8	0	0	0	28	2016-06-07 17:23:17
Insilico	0	8	8	0	8	0	0	0	798	2016-06-07 18:21:51
MAMBA	0	8	8	0	8	0	0	0	32	2016-06-07 17:23:35
MPACT_Drivers	0	8	8	0	8	0	0	0	40	2016-06-07 18:57:38
MPACT_exe	0	8	8	0	3	5	0	0	1639	2016-06-07 19:06:51
MPACT_libs	0	8	8	0	8	0	0	0	1064	2016-06-07 18:47:34
Tiamat	0	8	8	0	6	2	0	0	47	2016-06-07 19:26:05
TriBITS	0	8	8	0	0	8	0	0	1632	2016-06-07 17:19:05
VeraAPI	0	8	8	0	8	0	0	2	270	2016-06-07 19:24:31
VERAIn	0	8	8	0	8	0	0	0	1064	2016-06-07 17:55:35
VERAout	0	8	8	0	8	0	0	0	8	2016-06-07 17:58:18
VUQCore	0	8	8	0	8	0	0	0	32	2016-06-07 19:35:49
VUQDemos	0	7	7	0	7	0	0	0	2	2016-06-07 19:36:29

Post-Push Testing: TRIBITS_CTEST_DRIVER()



36

My CDash All Dashboards Log Out VERA Monday, May 05 2014 20:48:48 EDT

CASL

Dashboard Calendar Previous Current Next Project Settings

No update data as of Sunday, April 06 2014 - 23:00 EDT

Show Filters Advanced View Auto-refresh Help

Nightly

Site	Build Name	Update	Configure		Build		Test			Build Time	Labels
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_RELEASE_GCC461	0	0	74	0	60	0	1	625	Apr 07, 2014 - 01:11 EDT	(13 labels)
james007.ornl.gov	Linux-GCC-4.6.1-MPI_RELEASE_GCC461_WEEKLY	0	0	74	0	60	0	1	634	Apr 07, 2014 - 01:15 EDT	(13 labels)
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461	0	0	74	0	61	0	0	626	Apr 07, 2014 - 01:11 EDT	(13 labels)

Continuous

Site	Build Name	Update	Configure		Build		Test			Build Time	Labels
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461_CI	0	0	15	0	1	0	0	320	Apr 07, 2014 - 21:38 EDT	(2 labels)
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461_CI	0	0	15	6	1	0	0	53	Apr 07, 2014 - 19:14 EDT	(2 labels)
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461_CI	0	0	44	0	120	0	0	594	Apr 07, 2014 - 04:06 EDT	(11 labels)

VERA CDash Dashboard for 4/6/2014

- Collapsed summaries for each build case
- Nightly, CI, Experimental build cases

My CDash All Dashboards Log Out

Monday, May 05 2014 20:48:52 EDT

CDASH

VERA

DashboardCalendarPreviousCurrentNextProjectSettings

To update data as of Sunday, April 06 2014 - 23:00 EDT

Show FiltersAdvanced ViewAuto-refreshHelp

Continuous

Site	Build Name	Update	Configure		Build		Test			Build Time	Labels
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461_CI	0	0	13	0	1 ⁺¹⁴ ₋₁₄	0	0	53 ₋₂₇	Apr 07, 2014 - 19:18 EDT	VRIPSS
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461_CI	0	0	2	6 ⁺⁶	0				Apr 07, 2014 - 19:14 EDT	COBRA_TF

VERA CDash CI Iterations

- Individual packages built in sequence
 - Targeted emails for failed package build & tests
 - Failed packages disabled in downstream packages
- => Don't propagate failures!

My CDash All Dashboards Log Out

Monday, May 05 2014 20:51:59 EDT

CASL

VERA

Dashboard

Calendar

Previous

Current

Next

Project

Settings

No update data as of Sunday, April 06 2014 - 23:00 EDT

Show Filters

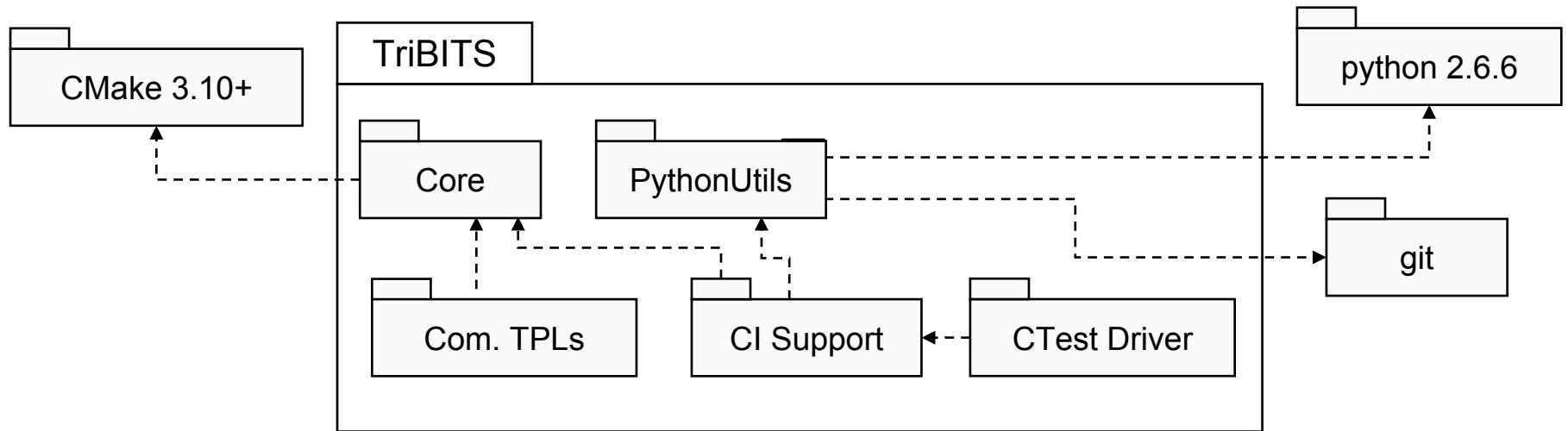
Advanced View

Auto-refresh

Help

Continuous

Site	Build Name	Update	Configure		Build		Test			Build Time	Labels
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461_CI	0	0	13	0	1 ⁺⁶ ₋₁₄	0	0	100 ⁺⁴⁷	Apr 07, 2014 - 21:45 EDT	VRIPSS
pu241.ornl.gov	Linux-GCC-4.6.1-MPI_DEBUG_GCC461_CI	0	0	2	0 ₋₆	0 ⁺¹²	0	0	220 ⁺²²⁰	Apr 07, 2014 - 21:38 EDT	COBRA_T



- **TriBITS Partitioning and Dependencies:**
 - **TriBITS Core** ([tribits/core/](https://tribits.org/core/)): Core TriBITS package-based architecture for CMake projects includes configure, build, test, install, deploy (tarballs) for multi-repo projects. (1M size)
 - **TriBITS Python Utils** ([tribits/python_utils/](https://tribits.org/python_utils/)): Some basic Python utilities that are not specific to TriBITS (e.g. [gitdist](#), [snapshot_dir.py](#)).
 - **TriBITS CI Support** ([tribits/ci_support/](https://tribits.org/ci_support/)): Support code for pre-push continuous integration testing (e.g. [checkin-test.py](#)).
 - **TriBITS CTest Driver** ([tribits/ctest_driver/](https://tribits.org/ctest_driver/)): Support for package-by-package testing driven by CTest submitting to CDash (e.g. [TribitsCTestDriverCore.cmake](#)).
- **Upcoming Work:**
 - Collect requirements for next generation of TriBITS
 - Upgrade to use modern target-based CMake implementation and features
 - More flexibility to pre-build packages and linking in as external packages (TPLs)

THE END

- **Contact:** rabartl@sandia.gov
- **Sponsors:**
 - CASL: Consortium for the Advanced Simulation of Lightwater reactors