

# TriBITS Developers Guide

**Author:** Roscoe A. Bartlett ([bartlettra@ornl.gov](mailto:bartlettra@ornl.gov))

## Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Structure of a TriBITS Project</b>	<b>1</b>
<b>3 References</b>	<b>1</b>
<b>4 Appendix</b>	<b>1</b>
4.1 History of TriBITS . . . . .	1

## 1 Introduction

This document describes the usage of the TriBITS (Tribal Build, Integration, Test System) to develop software projects. An initial overview of TriBITS is provided in the [TriBITS Overview](#) document which contains the big picture and provides a high-level road map to what TriBITS provides. This particular document, however, describes the details on how to use the TriBITS system to create a CMake build system for a set of compiled software packages.

## 2 Structure of a TriBITS Project

???

## 3 References

## 4 Appendix

### 4.1 History of TriBITS

TriBITS started development in November 2007 as a set of helper macros to provide a CMake build system for a small subset of packages in Trilinos. The initial goal was to just to support a native Windows build (using Visual C++) to compile and install

[SCALE] <http://scale.ornl.gov/>

these few Trilinos packages on Windows for usage by another project (the Sandia Titan project which included VTK). At that time, Trilinos was using a highly customized autotools build system. Initially, this CMake system was just a set of macros to streamline creating executables and tests. Some of the conventions started in that early effort (e.g. naming conventions of variables and macros where functions use upper case like old FORTRAN and variables are mixed case) were continued in later efforts and are reflected in the current. Then, starting in early 2008, a more detailed evaluation was performed to see if Trilinos should switch over to CMake as the default (and soon only) supported build and test system (see “Why CMake?” in [TriBITS Overview](#)). This led to the initial implementation of a scalable package-based architecture (PackageArch) for the Trilinos CMake project in late 2008. This Trilinos CMake PackageArch system evolved over the next few years with development in the system slowing into 2010. This Trilinos CMake build system was then adopted as the build infrastructure for the CASL VERA effort in 2011 where CASL VERA packages were treated as add-on Trilinos packages (see Section ???). Over the next year, there was significant development of the system to support larger multi-repo projects in support of CASL VERA. That led to the decision to formally generalize the Trilinos CMake PackageArch build system outside of Trilinos and the name TriBITS was formally adopted in November 2011. Work to refactor the Trilinos CMake system into a general reusable stand-alone CMake-based build system started in October 2011 and an initial implementation was complete in December 2011 when it was used for the CASL VERA build system. In early 2012, the ORNL CASL-related projects Denovo and SCALE [[SCALE](#)] adopted TriBITS as their native development build systems. Shortly after TriBITS was adopted the native build system for the the CASL-related University of Michigan code MPACT [???]. In addition to being used in CASL, all of these codes also have a significant life outside of CASL. Because they used the same TriBITS build system, it proved relatively easy to keep these various codes integrated together in the CASL VERA code meta-build. At the same time, TriBITS well served the independent development teams and non-CASL projects independent from CASL VERA. Since this, TriBITS was further extended and refined, driven by CASL VERA development and expansion. Independently, an early version of TriBITS from 2012 was adopted by the LiveV project footnote{<https://github.com/lifev/cmake>} which was forked and extended independently.