



Collaborative Software Development



Collaborative Software Development

Introduction:

CDash is an open source, web-based software testing server usable to collect and organize informations from a community of building clients.

CDash receive client informations as standard XML files. Once parsed, data is stored using a DBMS.

CDash automatically aggregates and analyzes received data. Results are categorized by different aspects (i.e. experimental or nightly builds, etc). From analysis results, different actions can be automatically performed

CDash also provides a web interface to display projects data



CDash main features [1/2]:

CDash implements a Dart-compliant project dashboard

- Collect informations regarding the setup of building systems (hardware and operating system configuration)
- Gathers results from configuration/compile/link phase of the build process (warnings, errors and output of programs)
- Stores build related statistical information (build time, etc.)
- When used in conjunction with automatic software testing tools, test results can be added to project dashboard (with tracking of tests coverage)
- Developers or project administrators can be alerted by email when a client fail building or testing



CDash main features [2/2]:

CDash can be tightly integrated with CMake and CTest

- Managing build process with CMake allow automatic data collection and communication with CDash server
- As with CMake, using CTest for unattended software testing allow automating management of data collection and communication.
(Moreover CMake can take care of CTest configuration)

CDash can also interact with other external tools

- Dynamic memory allocation can be checked with Valgrind or Purify
- Bug tracking can be managed through Mantis
- Automatic software documentation is managed with doxygen
- Support ViewCVS, Trac, Fisheye, ViewVC, WebSVN and CVSTrac to access software repositories



Collaborative Software Development

CDash dashboard example [1/3]:

Nightly													
Site	Build Name	Update		Configure			Build			Test			
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min
yellowstone.kitware	CDash-1.8-MySQL	0	0.1	0	0	0				0	0	76	5.1
yellowstone.kitware	CDash-1.8-PgSQL	0	0.1	0	0	0				0	1	75	8.7
yellowstone.kitware	CDash-SVN-MySQL	0	0.1	0	0	0				0	0	76	7
yellowstone.kitware	CDash-SVN-PgSQL	0	0.1	0	0	0				0	1	75	9.9
dashmacmini4.kitware	Release-1-6-MacOSX-xampp-1.7.2a	0	0	0	0	0.1	0	0	0.5	0	0	42	0.5
dash19.kitware	Release-1-6-Win32-xampp-1.7.3	0	0.1	0	0	0.1	0	0	34.9	0	79	18	34.9
dashmacmini4.kitware	trunk-MacOSX-xampp-1.7.2a-coverage	0	0	0	0	0				0	0	114	16.7



CDash dashboard example [2/3]:

Site: [yellowstone.kitware](http://yellowstone.kitware.com)

Processor Speed: 2.21GHz
64 Bits: 0
Processor Vendor: GenuineIntel
Processor Vendor ID: Intel Corporation
Processor Family ID: 6
Processor Cache Size: 2048
Number of logical CPUs: 1
Number of physical CPUs: 1
Number of logical CPU per Physical CPUs: 1
Total Virtual Memory: 2.33GB
Total Physical Memory: 1.97GB
Description: NA

Claimed by: Julien François

IP address: 24.199.146.138

Build name: [CDash-1.8-MySQL](#)

Site Name: [yellowstone.kitware](http://yellowstone.kitware.com)
Build Name: CDash-1.8-MySQL
Stamp: 20101211-0100-Nightly ([related builds](#))
Time: 2010-12-10T22:00:06 EST
Type: Nightly

OS Name: Linux
OS Platform: i686
OS Release: 2.6.31-22-generic
OS Version: #67-Ubuntu SMP Sat Oct 16 19:10:07 UTC 2010
Compiler Version: unknown

Current Build			Previous Build		
Stage	Errors	Warnings	Stage	Errors	Warnings
Update	0	0	Update	0	0
Configure	0	0	Configure	0	0
Build	0	0	Build	0	0
Test	0	0	Test	0	0



CDash dashboard example [3/3]:

Test coverage and dynamic memory allocation analysis

Coverage				
Site	Build Name	Percentage	LOC Tested	LOC Untested
dash17.kitware	Linux-g++4.0	75.38%	22920	7484
dash16.kitware	Linux-g++4.3	72.96%	27431	10165
hythloth.kitware	Linux64-bullseye-cov	86.4%	3234	509
dash22.kitware	Win32-vs9-Release-Coverage	74.15%	3244	1131

Dynamic Analysis			
Site	Build Name	Checker	Defect Count
dash17.kitware	Linux-g++4.0	Valgrind	0
FarAway.kitware	Linux-valgrind2	Valgrind	0



CDash Environment:

CDash is written in PHP

- It requires PHP \geq 5.3
- It requires an Apache web server

Two database backend available:

- MySQL \geq 5
- PostgreSQL \geq 8.3

Other PHP requirements:

- XSL (eXtensible Stylesheet Language) module/support enabled
- cURL module/support enabled
- GD module/support enabled



Current Status and Future plan:

Testing installation available at <http://lxkopio.pd.infn.it:3333/CDash>

- CDash 1.6.4
- Web server: Apache 2.2.8-3
- PHP 5.3.3 with cURL, GD and XSL built-in
- DBMS: MySQL 5.1.53-1

FastSim project dashboard available

At present we are working on the integration of the suite CMake-CTest-CDash.

- CMake introduction is planned starting from FastSim dev. version 0.2.7
- We plan to introduce CTest-CDash from version 0.2.8



Thanks

For your attention