cetbuildtools - Feature #10549

non-qualified products

10/16/2015 04:13 PM - Ron Rechenmacher

Status:	Accepted	**Start date:**	10/20/2015
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	33%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	2.50 hours

Description

We would like to build a flavored, non-qualified product in MRB.
I think that means cetbuildtools needs to support it.
We've emailed a 1 line patch to Lynn which may solve the issue.

Subtasks:

Feature # 10597: allow cetbuildtools to produce a product without a debug or prof quali...	**Assigned**
Feature # 10598: use the native compiler with cetbuildtools	**Accepted**
Feature # 10599: rename the simple option	**Closed**

History

#1 - 10/19/2015 09:23 AM - Lynn Garren

A bit more information from Ron:

More info about our use case is that we want mrb to be able to build TRACE
because another product we have (pcie_linux_kernel_module) depends on the TRACE kernel
module part of TRACE and we want to make sure we have a matching version.

Currently, the module part (which, of course, does not
interact with user space directly), needs to use the system compiler.
The lib and utility parts of the product use thread local storage and
atomic which is a relatively new features of C/C++ and should
not use the native compiler (on SLF6).
On SLF7, when it becomes available (very soon?), everything can use the native compiler.

#2 - 10/19/2015 09:31 AM - Lynn Garren

It should be noted that the setup_for_devolopment.noarch template is specifically designed for NULL flavored products and the rest of the infrastructure in mrb and cetbuildtools associated with "simple" is designed to support a NULL flavored product.

There would appear to be several different concepts here.

1. NULL flavored (nothing is compiled)
2. there are no qualifiers
3. use the native compiler

It is worth noting that the art team specifically requires a qualifier on all products which are built with a non-native compiler. cetbuildtools conforms to this requirement.

#3 - 10/19/2015 09:47 AM - Lynn Garren

Hum, there is a fourth idea: building with a non-native compiler, but without specifying either debug or prof.

#4 - 10/19/2015 10:15 AM - Eric Flumerfelt

The fourth option definitely sounds good for the library part of TRACE. We can play tricks in CMakeLists.txt to use the native compiler for the kernel module which requires it.

For right now, we can make a "e7:debug" build which would be used in all e7-based configurations, but it would be best to have a simple "e7" qualifier from our cetbuildtools-based build.

We'll deal with SLF7 when we get there.

#5 - 10/19/2015 10:34 AM - Lynn Garren

I am concerned by the idea that you are using two different compilers within the same build. None of the infrastructure is designed to support that.

#6 - 11/02/2015 11:34 AM - Lynn Garren

- Status changed from New to Accepted