

Morph-M - Anomalie #21

Goes 64 bits sous windows Boost 1.34.1

12/01/2008 10:08 AM - Serge Koudoro

Status:	Nouveau	Start date:	09/17/2007
Priority:	Bas	Due date:	
Assignee:		% Done:	30%
Category:	old plone Bugs	Estimated time:	0.00 hour
Target version:			

Description

Goes 64 bits sous windows Boost 1.34.1

Description:

Les solutions de Visual 2005 ont été mises a jour pour la compilation de MorphM en 64 bits sous Windows
La version de boost est la 1.34.1
Lors de la compilation en debug, boost met est en place des tests de concepts (cf.http://www.boost.org/libs/concept_check/concept_check.htm). C'est une bonne méthode pour contraindre la programmation de fonctions/classes template
Un de ces tests est déclenché dans commonGraph quand ont veut utiliser l'algorithme de MST de Boost
Car.....
Windows et les autres plateformes ne définissent pas les long et unsigned long de la meme facon
Ce qui fait que Boost déclence un asert quand il voit passé un __int64 ou un unsigned __int64

#1 17/09/2007 14:34 (Thomas)

Change: status: "pending" -> "accepted"
Change: assignees: "[]" -> "[faessel, 'morphmdev', 'noyel']"
Change: topic: "" -> "Others"
Change: solution: "" -> "Bon ben faut ajouter un petit bout de code dans boost/concept_check.hpp
ligne 149
""""

```
template <class T>
struct IntegerConcept {
void constraints() {
#if !defined BOOST_NO_TEMPLATE_PARTIAL_SPECIALIZATION
x.error_type_must_be_an_integer_type();
#endif
}
}
T x;
};
#if !defined BOOST_NO_TEMPLATE_PARTIAL_SPECIALIZATION
template <> struct IntegerConcept<short> { void constraints() {} };
template <> struct IntegerConcept<unsigned short> { void constraints() {} };
template <> struct IntegerConcept<int> { void constraints() {} };
template <> struct IntegerConcept<unsigned int> { void constraints() {} };
template <> struct IntegerConcept<long> { void constraints() {} };
template <> struct IntegerConcept<unsigned long> { void constraints() {} };
// etc.
```

- ICI

```
#ifdef WIN64
template <> struct IntegerConcept<__int64> { void constraints() {} };
template <> struct IntegerConcept<unsigned __int64> { void constraints() {} };
#endif *****
#endif
```

""

On notera le etc... qui n'est pas de moi

il se peut que ce soit corriger dans le CVS de boost mais pas dans la version en téléchargement pour le moment"

Change: importance: "medium" -> "low"

Change: title: "" -> "Goes 64 bits sous windows Boost 1.34.1"

Change: classification: "Bug" -> "Bug+Solution"

Change: description: "" -> "Les solutions de Visual 2005 ont été mises a jour pour la compilation de MorphM en 64 bits sous Windows

La version de boost est la 1.34.1

Lors de la compilation en debug, boost met est en place des tests de concepts

(cf. http://www.boost.org/libs/concept_check/concept_check.htm). C'est une bonne méthode pour contraindre la programmation de fonctions/classes template

Un de ces tests est déclenché dans commonGraph quand ont veut utiliser l'algorithme de MST de Boost

Car.....

Windows et les autres plateformes ne définissent pas les long et unsigned long de la meme facon

Ce qui fait que Boost déclence un asert quand il voit passé un __int64 ou un unsigned __int64"

History

#1 - 12/01/2008 10:10 AM - Serge Koudoro

- Category set to old plone Bugs

- Priority changed from Normal to Bas

- Start date set to 09/17/2007

- % Done changed from 0 to 30

#2 - 12/01/2008 10:10 AM - Serge Koudoro

#2 comment 18/09/2007 14:28 (Thomas)

Comment:

Suite a un poste de ma part sur la mailing de boost, un patch a été appliqué

cf <http://svn.boost.org/trac/boost/changeset/39363>

Wait and see

#3 - 12/01/2008 10:20 AM - Serge Koudoro

#4 04/10/2007 11:56 (Thomas)

Change: status: "accepted" -> "wont_fix"

Comment:

Quelques remarques en plus sur le dev 64

Bon ben morphm en debug 64 c'est pas gagné car cf ici

<http://connect.microsoft.com/VisualStudio/feedback/ViewFeedback.aspx?FeedbackID=99655>

MorphM est compilé sans Safe STI (voire épisode précédent)

MAIS et je pense boost aussi link avec msvc80d.dll qui est compilée avec

_HAS_ITERATOR_DEBUGGING=1 (IE Safe)

Dixit le post c'est corrigé dans le SP1

Or le SP1 je l'ai installée....

Plus rigolo encore

cf

<http://blogs.msdn.com/vcblog/archive/2007/02/26/sti-destroyer-of-bugs.aspx>

""

However, everything has bugs, including bug-finding machinery. So far I've

worked on a couple such bugs. One of them, a regression in Visual Studio 2005

SP1, was triggered by compiling a program in debug mode, with iterator debugging

disabled (which is not the default), obtaining an iterator into a container, and

destroying the container before the iterator. The reason why this

triggers a
crash is long and complicated, and figuring out the history was
fun. We've fixed
this regression in Orcas.

En résumé et si je ne dis pas de bêtise
ON NE PEUT PAS FAIRE les librairies en DEBUG SANS la safe STL
Et si on lit bien la fin du post

The reason why this triggers a crash is long and complicated, and
figuring out
the history was fun. We've fixed this regression in Orcas.

On a qu'a acheter Orcas ie Visual 2008
En gros le bug n'est pas fixé et ne sera pas fixé dans Visual 2005
dixit les gens de crosoft

Bref....

Ceci termine les pbs de support "experimental" sous Visual 2005
Logiquement on a des Releases 32 et 64 fonctionnelles
Pour le debug on repassera