

Morph-M - Anomalie #32

Ambiguïté de Compilation du a la grande souplesse du Cvariant

12/01/2008 02:29 PM - Serge Koudoro

Status:	Nouveau	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	old plone Bugs	Estimated time:	0.00 hour
Target version:			

Description

Ambiguïté de Compilation du a la grande souplesse du Cvariant

Description:

Problème découvert par raffi et je pense qu'il est important de laisser une trace:

Exemple:

int operator_less Pixel_3 ceci est invalide et ne devrait jamais compiler

int-> Cvariant

Pixel_3-> Cvariant

operator_less -> Cvariant EXISTE!! rajouter pour pqlInterface

Donc ça compile et ca fait une erreur dormante

Pire: je ni connais rien en compilateur mais si un compilo a deux chemins de compilation valide:

int < int ou Cvariant(int) lessoperatorCvariant CVariant(int) prend il toujours le chemin le plus court????

Truc ou je comprend plus :

TOut operationBuilder(const CVariant& cv1, const CVariant&cv2) l.667 de CommonVariant.cpp n'est censé autoriser les opérations que pour les scalaires donc la je suis perdue.

Donc si le bug est déjà résolue on vire tout de suite mon message inutile et je retourne me caché.

Sinon y a des questions a se poser.

#4 12/11/2007 09:58 (Tibs)

Change: status: "pending" -> "deferred"

Comment:

Bon ben moi j'y comprend rien, comme je crois pas avoir vu de catastrophe à ce sujet dernièrement. C'est le moment de tester l'option "Différer" du collecteur de bug ;)

#3 comment 06/09/2006 12:02 (Anonymous User)

Comment:

> si un compilo a deux chemins de compilation valide:

> prend il toujours le chemin le plus court????

Heureusement, oui. Les règles définissant l'ordre des chemins sont à lire ici:

<http://www.csci.csusb.edu/dick/c++std/cd2/over.html#over.ics.rank>

(je vous invite d'ailleurs à retenir ce lien et à vous y référer pour vos interrogations futures sur le comportement de C++, je ne serai pas toujours là pour calmer vos angoisses...)

Il faut noter au passage qu'il ne s'agit pas d'une ambiguïté au sens standard du terme (i.e. au moins deux chemins possibles équivalents).

Pour moi ceci n'est pas un bug, car le code plante proprement dans l'opérateur de comparaison.

Mais si ça vous gêne vraiment, vous avez le source.

#2 06/09/2006 11:26 (Thomas)

Change: topic: "UI" -> "Others"

#1 06/09/2006 11:24 (Thomas)

Change: topic: "" -> "UI"

Change: solution: "" -> "Solution pas du tout élégante:

Ne pas faire de surcharges des opérateurs pour éviter ambiguïté de compilation.

Les remplacer par des functor aux noms explicite less_Cvariant,....

Oui c'est moche"

Change: title: "" -> "Ambiguïté de Compilation du a la grande souplesse du Cvariant"

Change: description: "" -> "Problème découvert par raffi et je pense qu'il est important de laisser une trace:

Exemple:

int operator_less Pixel_3 ceci est invalide et ne devrait jamais compiler

int-> Cvariant

Pixel_3-> Cvariant

operator_less -> Cvariant EXISTE!! rajouter pour pqlInterface

Donc ça compile et ca fait une erreur dormante

Pire: je ni connais rien en compilateur mais si un compilo a deux chemins de compilation valide:

int < int ou Cvariant(int) lessoperatorCvariant Cvariant(int) prend il toujours le chemin le plus court????

Truc ou je comprend plus :

TOut operationBuilder(const Cvariant& cv1, const Cvariant&cv2) l.667 de CommonVariant.cpp n'est censé autoriser les opérations que pour les scalaires

donc la je suis perdue.

Donc si le bug est déjà résolue on vire tout de suite mon message inutile et je retourne me caché.

Sinon y a des questions a se poser.

"

History

#1 - 12/01/2008 02:30 PM - Serge Koudoro

- Category set to old plone Bugs