Subject: Re: Compound New <:PACKAGE:> name etc. [FEATURE REQUEST]
Posted by gprentice on Sat, 20 May 2006 11:37:51 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Sat, 20 May 2006 09:35gprentice wrote on Fri, 19 May 2006 12:23I'm not sure I understand but it seems the purpose of the author/prefix/version items is
1. to make it easier to duplicate the package in future, however I don't understand why this makes it easier ??
2. Is your "duplicate package" command going to go through all the package source files and change #include <packagenameV1/xxx.h> to #include <packagenameV2/xxx.h> ??

1. Yes. And easier because of:
2a. First of all, try duplicate (your usual "copy directory" way) not HelloWorld but packages which have *.iml and *.lay. Tell your experience!

2b. Imagine you work with a team. Try to produce 2 different packages from 1 template... Or in 2 steps create 2 packages (a package which uses the other package which will be also versioned (with *.iml and *.lay). And test quickly 2*2 combinations. Experiences?
Or imagine creating a step-by-step tutorial-in-action... with different subtopics and combinations...
Edit2: Or having your testable and combinable (!) your code snippets database...

2c. And because the trick (remember, we were talking about)

 #define IMAGEFILE   PACKAGE_DIR/Vega.iml>

doesn't work well with GCC and BLITZ...

2d. And/or create a template which simply produces
#include ForlanoVega1/Vega.h etc...
customizable and shows them in the dialogue each time you enter a char!..

2e. And/or updates those names according to template params...

Edit:
2. Yes. and *.upp

Ok, so layout files are a problem because they include path information in the LAYOUTFILE #define - but it's not hard to search for all LAYOUTFILE in the package and change the pathname. Similarly with #includes that include the package name - they can be changed very quickly with a regular expression and global replace over all package files, but it seems you're duplicating packages so frequently this probably doesn't appeal to you

Unfortunately, the mechanism you've used, which the uninitiated would, not unreasonably, expect to work, probably doesn't work portably

Graeme