

Anubis - Feature #20
New keyword: obsolete

09/30/2007 04:11 PM - Anonymous

Status:	Assigned	Start date:	
Priority:	Low	Due date:	
Assignee:	Alain Prouté	% Done:	0%
Category:	Compiler	Estimated time:	0.00 hour
Target version:	1.x	Triage Stage:	
Platform:			
Resolution:			
Description			
<p>As standard library and functions or types that compose it aren't yet perfectly well designed, and name changes can be frequent, adding a keyword 'obsolete' (or any thing like this) could be a great help.</p> <p>This keyword does nothing else than producing a warning during compile time, just to remember to not use this function.</p> <p>A variation of this could be the keyword : obsolete(new_function)</p> <p>With this, the compiler can produce a very useful warning, telling which function to use instead of this one.</p> <p>The keyword could be inserted into function or type definition like this :</p> <pre>public define obsolete(new_function_name) [[ReturnType]] old_function_name (...) = <function body>.</pre> <p>Or maybe better (as we can suppose that 'obsolete' functions or type are always public ones):</p> <pre>obsolete(new_function_name) define [[ReturnType]] old_function_name (...) = <function body>.</pre> <p>which made automatically this function <i>public</i> but <i>obsolete</i>.</p> <p>Private function can always be easily renamed, so there is no need to mark them as obsolete.</p>			

History

#1 - 09/30/2007 07:43 PM - Alain Prouté
- Status changed from New to Assigned

Good idea. I suggest this for the syntax:

obsolete "... compiler message ..."
public define ...

obsolete "... compiler message ..."
public type ...

The message will be sent (only once) if the paragraph is used.

We will also have to consider the case of some primitive obsolete features like Int32 which are not defined by paragraphs.