U++ - Bug #728
ide: Advanced/Format the code in editor freezes

03/21/2014 11:38 AM - Miroslav Fidler

Status:	Approved	**Start date:**	03/21/2014
Priority:	Normal	**Due date:**	
Assignee:	Massimo Del Fedele	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

with really long line:

bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 1) return false; __Expand1(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1) { return SplitTo(s, delim, false, p1); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 1) return false; __Expand1(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1) { return SplitTo(s, chr, false, p1); } bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1, String& p2) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 2) return false; __Expand2(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1, String& p2) { return SplitTo(s, delim, false, p1, p2); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1, String& p2) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 2) return false; __Expand2(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1, String& p2) { return SplitTo(s, chr, false, p1, p2); } bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1, String& p2, String& p3) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 3) return false; __Expand3(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1, String& p2, String& p3) { return SplitTo(s, delim, false, p1, p2, p3); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1, String& p2, String& p3) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 3) return false; __Expand3(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1, String& p2, String& p3) { return SplitTo(s, chr, false, p1, p2, p3); } bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 4) return false; __Expand4(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1, String& p2, String& p3, String& p4) { return SplitTo(s, delim, false, p1, p2, p3, p4); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 4) return false; __Expand4(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1, String& p2, String& p3, String& p4) { return SplitTo(s, chr, false, p1, p2, p3, p4); } bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 5) return false; __Expand5(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1, String& p2, String& p3, String& p4, String& p5) { return SplitTo(s, delim, false, p1, p2, p3, p4, p5); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 5) return false; __Expand5(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1, String& p2, String& p3, String& p4, String& p5) { return SplitTo(s, chr, false, p1, p2, p3, p4, p5); } bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 6) return false; __Expand6(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6) { return SplitTo(s, delim, false, p1, p2, p3, p4, p5, p6); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 6) return false; __Expand6(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6) { return SplitTo(s, chr, false, p1, p2, p3, p4, p5, p6); } bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 7) return false; __Expand7(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7) { return SplitTo(s, delim, false, p1, p2, p3, p4, p5, p6, p7); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 7) return false; __Expand7(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7) { return SplitTo(s, chr, false, p1, p2, p3, p4, p5, p6, p7); }

bool SplitTo(const char *s, int delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7, String& p8) { Vector<String> r = Split(s, delim, ignoreempty); if(r.GetCount() < 8) return false; __Expand8(E__NFIf) return true; } bool SplitTo(const char *s, int delim, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7, String& p8) { return SplitTo(s, delim, false, p1, p2, p3, p4, p5, p6, p7, p8); } bool SplitTo(const char *s, const char *delim, bool ignoreempty, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7, String& p8) { Vector<String> r = Split(s, chr, ignoreempty); if(r.GetCount() < 8) return false; __Expand8(E__NFIf) return true; } bool SplitTo(const char *s, const char *delim, String& p1, String& p2, String& p3, String& p4, String& p5, String& p6, String& p7, String& p8) { return SplitTo(s, chr, false, p1, p2, p3, p4, p5, p6, p7, p8); }

This line pasted into file, selected, format invoked -> freeze

History

#1 - 03/21/2014 12:06 PM - Miroslav Fidler

- Status changed from New to Approved