

平成12年9月11日

PC版/Indy版 SHC/C++コンパイラ提出リスト

No.	バージョン	内 容	備 考
1	SHC/C++ V5.1R01	<ul style="list-style-type: none"> ・Cに// コメントをサポート ・コマンドライン長を128文字 -> 4096文字 ・ファイル名長を128文字 -> 251文字 ・#pragma aligndata32をサポート ・環境変数 SHC_INCにパス名区切り"."を追加 ・コピーライト表示オプションの追加 ・標準インクルードの2重インクルード対応 ・-1.0の定数を命令から生成 ・文字列長を512文字 -> 32767文字 	Indy版提供
2	SHC/C++ V5.1R03	<ul style="list-style-type: none"> ・FMAC命令使用時、4099エラー ・配列初期値が大量の文字列でmemory overflow ・C++ char* cp=&(* (char *)0x01);で4099エラー ・複雑なパラメタ式の組み込み関数でobject不正 ・setjmp で FR3 と FR5 の退避順序が逆 ・setjmp_a で FR レジスタの退避領域が不正 ・C++ boolをサポートしました。 ・enumメンバの最後の","(カンマ)をエラーから除外しました。 ・浮動小数点初期値の桁数が多い場合をエラーから警告に変更しました。 ・switch文のコード展開方式をコンパイルオプションで指定できるようにしました。(if then 方式、テーブル方式) ・ビットフィールドの型にenumを許すようにしました。 	
3	SHC/C++ V5.1R04	<ul style="list-style-type: none"> ・FMACコード展開時のレジスタ割り付け不正 ・C++構造体代入時のコード不正 	
4	SHC/C++ V5.1R05	<ul style="list-style-type: none"> ・帰納変数削除最適化不正 ・クラス配列のnewでメモリ不足の場合実行時エラー ・C++で構造体のサイズ不正 ・C++でキャスト付きの初期値不正 ・bool型のエンコード名不正 	
5	SHC/C++ V5.1R06	<ul style="list-style-type: none"> ・コピーサイズが65536以上でmemcpyライブラリの動作が不正 ・仮想基底クラスが2レベル以上の基底クラスを持つコンストラクタの動作不正 ・switch文本体のオブジェクトサイズが32KB以上のとき動作不正 	
6	SHC/C++ V5.1R08	<ul style="list-style-type: none"> ・関数引数の暗黙の型変換のときユーザ定義の 	Indy版提供

		copy constructorがCallされない ・bool型を含むEC++libraryのエンコード名不正	
7	SHC/C++ V5.1R09	<ul style="list-style-type: none"> ・最適化時、R0レジスタ割付不正 ・switch文内で不当にオブジェクトのデストラクタコールする場合がある。 ・continue文のあるfor, whileループで帰納変数削除の最適化不正となる場合がある ・不当に6203エラーを出力する ・tan(3/4)が、-0.0になる ・入力値が大きな数のときatan, atan2の値が不正 ・入力値が大きな数のときsin/cosの値が-1 ~ 1の範囲外の値になる ・組み込み関数ld_extで、delayed slot最適化不正となる場合がある 	
8	SHC/C++ V5.1R11	<ul style="list-style-type: none"> ・-case=tableオプションでif文展開する場合がある ・ループ内で最適化によって作成された後置++式がオブジェクト不正 ・ヘッダファイルmath.hとstdlib.hで宣言されているHUGE_VALの名前が一致しない ・デストラクタで明示的にreturn文を記述するとデストラクタがコールされない場合がある ・privateメンバを指定すると不当にエラーを表示する場合がある ・初期値にメンバ関数の属するクラスのメンバを指定するとインターナルエラーが出力される場合がある 	
9	SHC/C++ V5.1R12	・if 文等の条件式の直後のブロックが空で、かつ条件式が定数との判定である場合にインターナルエラーが出力される	
10	SHC/C++V5.1R13	<ul style="list-style-type: none"> ・C++ で _errno, HUGE_VAL が Multiple Linkage-specificationになるケースがある ・FMACを使用するコードでFR0を壊した不正なオブジェクトを生成するケースがある ・外部変数の宣言がなくてもコンパイルエラーにならないケースがある 	

以上