

VisualDSP++ 4.0 (Updated September 2006) Release Note

The following release note concerns the September 2006 Update to the VisualDSP++ 4.5 release. This release is inclusive of previous Updates. The contents of future Updates will be inclusive of all previous Updates. The release notes for past Updates are appended to the end of this release note.

Identifying Which Update Is Currently Installed on Your System

The Update level is identified in three places:

1. The Add/Remove Programs Control Panel entry for VisualDSP++ 4.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file ...\\System\\VisualDSP.ini, in the ProductName key.

Installing an Update

The procedure for installing an Update to VisualDSP++ is described below. Note that with VisualDSP++'s support for installing multiple instances of itself, it is possible to "trial" an Update in a new directory before switching over your "golden" tools installation to the Update.

1. Use the Start Menu to navigate to VisualDSP++'s "Maintain this Installation" item.
2. Select "Go to the Analog Devices website" and click "Next". This will launch your web browser and navigate it to the proper URL to download Updates from.
3. Download the VisualDSP++ Update file (.VDU) of interest to your hard disk. Note that these files have a .VDU file extension and cannot be executed directly.
4. Navigate to "Maintain this Installation" again. If you have multiple installations of VisualDSP++ on your computer, be doubly sure you are navigating to the installation you wish to Update.
5. Select "Apply a downloaded Update" and click "Next". Click the "... " browser button and navigate to the .VDU file that you downloaded in step 3. Click "OK", then "Next".
6. Follow the on-screen prompts to complete the installation of the Update.

Significant Additions

The primary purpose of VisualDSP++ Updates is to address problems and stabilize the release. Significant new functionality is not expected to be introduced in an Update. However, incremental support (i.e., emulation, example programs, header files, default LDF, errata accommodations, EZ-KIT Lite software, etc.) for new semiconductor products will be added as these products become available and gain support within the VisualDSP++ tools.

In this release:

1. Software support for the ADSP-BF538F EZ-KIT Lite has been added. This includes debug connectivity via the on-board USB debug agent, the flash programmer (both GUI support and underlying drivers), and LwIP (Ethernet) drivers. Flash drivers are provided for both the on-chip flash found on the ADSP-BF538F and the off-chip flash device on the EZ-KIT Lite. The on-line help system has been updated to include this product.
2. The System Service Library (SSL) has been enhanced to support ADSP-BF538 Blackfin processor. Included in this Update is support for the EBIU, Dynamic Power Management, DMA, Interrupt, Deferred Callback, Timer, Flag and Port Control system services for the ADSP-BF538 processor. The default LDFs have been updated to link against SSL.
3. Blackfin device drivers have been updated. The adi_ad1836a_ii and adi_ad1938_ii codec drivers now support automatic SPORT configuration. PPI, UART, SPI, TWI and SPORT device drivers for the ADSP-BF538 processor have been introduced.

4. Software support for the ADSP-21375 EZ-KIT Lite has been added. This includes debug connectivity via the on-board USB debug agent, and well as flash programmer GUI support and underlying drivers. Execution from external memory is now supported in simulator, emulator, and EZ-KIT Lite debugging sessions. The on-line help system has been updated to include this product.

Changes to Existing Behaviors, Projects, and Source Code

When addressing problems, we attempt to make any changes backward compatible with existing projects. However, depending on the nature of a problem, compatibility issues are sometimes unavoidable. This section highlights any changes in the Update that may require the modification of “working” projects or otherwise influence existing behavior.

In this release:

1. Within the Blackfin DMA Manager’s include file, `adi_dma.h`, an additional field named `CallbackFlag` has been added to the data structures that describe large and small model descriptors, `ADI_DMA_DESCRIPTOR_LARGE` and `ADI_DMA_DESCRIPTOR_SMALL` respectively. This field should be set to `TRUE`, if a callback is requested after the descriptor has been processed or `FALSE` if no callback is requested after the descriptor has been processed. Previously, the `DI_EN` bit within the configuration register of the descriptor was used to trigger a callback.

This change affects only user code that explicitly calls the `adi_dma_Queue()` function.

2. Source code files that make calls into the System Services (code that includes the file “`services.h`”) should be rebuilt after installation of this update.

Problems Addressed

The following table is a list of the problems addressed in this Update. Details on any particular problem can be found on the Tools Anomaly web page. Note that after the Issues headings in the top half of the Tools Anomaly web page, problems are detailed in numeric order. The URL is:

<http://www.analog.com/processors/technicalSupport/toolsAnomalies.html>

Processor Family	Problem Number	Tool	Description
All	28661	Install	FYI Norton Internet Security prevents installation
All	28799	Run Time Libraries	fread may signal EOF prematurely
All	28849	Compiler	Compiler driver+LDF requires dummy.c file
All	28895	Run Time Libraries	C++ runtime support for alternative heaps fails to link (li1021)
All	28948	Compiler	loop pragmas don't work on C++ loops with embedded declarations
All	29012	VDK	VDK kThreadSwitched history events don't call UserHistoryLog
All	29090	IDDE	-g is added when convert project from 4.0 to 4.5 with -g>0 in th
All	29378	VDK	Incorrect behaviour following VDK timequeue wraparound
Blackfin	25362	Emulator	Emulator returns unknown family when targeting BF534 rev 0.2
Blackfin	26646	CRTGen	Rev 1.7 BF533 EZ-Kit not properly supported in generated CRT
Blackfin	28335	Simulator	C++ exceptions cannot be used on single core BF561 simulator
Blackfin	28339	Simulator	Simulator not updating registers correctly
Blackfin	28595	IDDE	Additional include directories not always added to command line
Blackfin	28764	System Services	1836A_ii codec device driver fails in TDM mode
Blackfin	28819	IDDE	Adding file to project with user defined config cause error
Blackfin	28823	Compiler	3-byte structs returned from functions incorrectly when -O used
Blackfin	28839	IDDE	impossible sclk value in Project wizard
Blackfin	28875	Run Time Libraries	Typo in cdefBF532.h - voidl instead of void
Blackfin	28967	Configurator	configurator doesn't handle more than 11 devices properly
Blackfin	29025	VDK	The placement of TMK and VDK libraries can cause link errors
Blackfin	29072	IDDE	Blackfin processors not listed in EL Global Properties dialog
Blackfin	29177	Run Time Libraries	SIC_IMASK set with bad bit before DMA transfer (meminit)
Blackfin	29231	Loader	BF533 rev 0.5 loader files add a zero byte for each data byte
SHARC	28857	Emulator	primes for 21061 does not run
SHARC	28882	Run Time Libraries	SIMD libdsp function vecdotf() might fail
TigerSHARC	28720	Compiler	TigerSHARC wrapper generation/regs_clobbered not saving K conds
TigerSHARC	28736	Splitter	The splitter counts one more byte per word in .stk format
TigerSHARC	28752	Run Time Libraries	TigerSHARC fread can incorrectly return 0 in byte-address mode
TigerSHARC	28907	VDK	TIMER0H register not set when Timer interrupt is set to None

TigerSHARC	29032	IDDE	Can't load Annotations
TigerSHARC	29043	Assembler	invalid warning about mult instruction option
TigerSHARC	29227	VDK	VDK API level check can cause false positive Kernel Panics

VisualDSP++ 4.5 (Updated July 2006) Release Note

The following release note concerns the July 2006 Update to the VisualDSP++ 4.5 release. This is the first in what is anticipated to be a series of Updates. The contents of future product Updates will be inclusive of all previous Updates. At that time, the release notes for past Updates will be appended to the end of the current release note.

Identifying Which Update Is Currently Installed on Your System

The Update level is identified in three places:

1. The Add/Remove Programs Control Panel entry for VisualDSP++ 4.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file ...\\System\\VisualDSP.ini, in the ProductName key.

Installing an Update

The procedure for installing an Update to VisualDSP++ is described below. Note that with VisualDSP++'s support for installing multiple instances of itself, it is possible to "trial" an Update in a new directory before switching over your "golden" tools installation to the Update.

1. Use the Start Menu to navigate to VisualDSP++'s "Maintain this Installation" item.
2. Select "Go to the Analog Devices website" and click "Next". This will launch your web browser and navigate it to the proper URL to download Updates from.
3. Download the VisualDSP++ Update file (.VDU) of interest to your hard disk. Note that these files have a .VDU file extension and cannot be executed directly.
4. Navigate to "Maintain this Installation" again. If you have multiple installations of VisualDSP++ on your computer, be doubly sure you are navigating to the installation you wish to Update.
5. Select "Apply a downloaded Update" and click "Next". Click the "... " browser button and navigate to the .VDU file that you downloaded in step 3. Click "OK", then "Next".
6. Follow the on-screen prompts to complete the installation of the Update.

Significant Additions

The primary purpose of VisualDSP++ Updates is to address problems and stabilize the release. Significant new functionality is not expected to be introduced in an Update. However, incremental support (i.e., emulation, example programs, header files, default LDF, errata accommodations, EZ-KIT Lite software, etc.) for new semiconductor products will be added as these products become available and gain support within the VisualDSP++ tools.

In this release:

1. The Blackfin compiler, runtime, VDK, and SSL libraries include new workarounds for hardware anomalies 05-00-0189 and 05-00-0283. The compiler will automatically enable these workarounds when building for parts and silicon revisions that require them. Alternatively they can be enabled using the `-workaround` switch.

05-00-0283 –

One part of the workaround is to include a code sequence in all event handlers. The sequence makes a mispredicted jump over a dummy MMR read. This must be done before any `SSYNCS` in the handler. This sequence is generated by the compiler for C/C++ based event handlers that use `#pragma interrupt` or `sys/exception.h` defined macros such as `EX_INTERRUPT_HANDLER`. The two handlers affected in the

runtime libraries are `_cplb_hdr` and the interrupt dispatcher `_despint` which have been modified to include the workaround.

The second part of the workaround is to avoid system MMR writes in the two instructions after a not-predicted conditional jump. The compiler will insert `nop` instructions to avoid this when it identifies the problem sequence.

These workarounds can be enabled using the `-workaround stalled-mmr-write-283` switch. When the workaround is enabled the macro `__WORKAROUND_STALLED_MMR_WRITE_283` is defined at compile, assemble and link stages

05-00-0198 –

A workaround for this anomaly was already available in the compiler. However the conditions which cause anomaly have changed to include a new code sequence. The compiler has been modified to identify this new sequence. The anomaly may occur where MMR reads or writes occur immediately after a stalled memory read. The compiler will avoid such code being generated for C/C++ compiled code. The runtime libraries are safe against this anomaly.

This workaround can be enabled using the `-workaround sdram-mmr-read` switch. When the workaround is enabled the macro `__WORKAROUND_SDRAM_MMR_READ` is defined at compile, assemble and link stages.

2. There has been a change of compiler behavior relating to MMR (Memory Mapped Register) accesses and volatile variables. The new switch `-no-assume-vols-are-mmrs` has been added.

There are various MMR related hardware errata that the compiler supports workarounds for; 05-00-0122, 05-00-0157, 05-00-0198, 05-00-0283. Previously the compiler would only implement these workaround for accesses that it could absolutely determine were to MMRs. This in practice meant that only literal MMR addresses accesses could be determined accurately. More complex accesses, for example using addresses stored in variables, might not be identifiable as MMR accesses and could therefore result in the various anomalies being hit.

The compiler has been modified to try and avoid missing these more complex MMR accesses. If there is an access to a variable that is defined as `volatile`, and the compiler cannot determine that the access is not to an MMR, the compiler will now assume it is an access to an MMR unless the new switch `-no-assume-vols-are-mmrs` is used.

Changes to Existing Behaviors, Projects, and Source Code

When addressing problems, we attempt to make any changes backward compatible with existing projects. However, depending on the nature of a problem, compatibility issues are sometimes unavoidable. This section highlights any changes in the Update that may require the modification of “working” projects or otherwise influence existing behavior.

In this release no changes have been identified.

Problems Addressed

The following table is a list of the problems addressed in this Update. Details on any particular problem can be found on the Tools Anomaly web page. The URL is:

<http://www.analog.com/processors/cda/epTASearch>

Processor Family	Reference Number	Tool	Description
All	28180	Compiler	modena test c0527101 fails byte addressing when compiled -eh
All	28225	Compiler	C++ exceptions thrown from inline virtual functions may fail
All	28244	VDK	Issue with dynamically created VDK components at startup
All	28271	Run Time Libraries	Increase in code size for printf
All	28341	Run Time Libraries	attributes missing in libx dojs
All	28399	Compiler	static C++ classes can cause bad debug
All	28876	Assembler	Cannot perform source-level debug of assembly source files
All	28929	Emulator	USB-ICE inoperable when updating to 4.5 while connected
All	28935	VDK	User's timer interrupt settings can be overwritten
All	29140	Emulator	RoHS USB-ICE does not work with base 4.5 install
Blackfin	28043	Loader	Loader supports different default Rev #s from what it should.
Blackfin	28229	IDDE	Annotations left in source pane after they are turned off
Blackfin	28287	Loader	Zero padding to booting stream
Blackfin	28297	Compiler	Compiler internal error (macdefs.c:1162) with -O
Blackfin	28305	Run Time Libraries	ftell() with -full-io in text mode can return incorrect position
Blackfin	28309	Compiler	Non-interrupt safe prologue code generated for BF535
Blackfin	28338	Compiler	INTERNAL COMPILER ERROR: No switch note found
Blackfin	28383	Assembler	.inc/binary produces corrupted doj
Blackfin	28410	Run Time Libraries	Cache flushing on BF535 and wireless parts doesn't work
Blackfin	28445	IDDE	If I add cplbtable and then disable cache project does not link
Blackfin	28450	IDDE	configurator screen not coming to front
Blackfin	28467	IDDE	errors and other issues removing configurations
Blackfin	28472	Run Time Libraries	Possibility of erroneous result computed by fir_decima_fr16()
Blackfin	28487	Run Time Libraries	Wrong comment in the source of the radix2 FFT library functions
Blackfin	28497	Run Time Libraries	Incorrect macro in defBF534.h and defBF538.h
Blackfin	28517	LDFGen	Possible link error with generated BF561 LDF and mem init
Blackfin	28521	LDF	OTHERCORE not implemented correctly in default multi-core LDFs
Blackfin	28588	Compiler	bad compare of unsigned short and unsigned literals -O
Blackfin	28600	Loader	Loader does not work with Rev 0.3 for 539.
Blackfin	28679	Loader	Remove the ignore block from loader files.
Blackfin	28686	LDFGen	Single core generated LDF uses \$OBJECTS before definition
Blackfin	28688	Run Time Libraries	Instance of speed path anomaly 05-00-0209 in cache flush func
Blackfin	28689	LDFGen	LDFGen does not always use the correct CPLB table
Blackfin	28710	Loader	Loader need to support Rev. 0.5 for 531/2/3
Blackfin	28765	IDDE	project not restored after starting connection-less IDDE
Blackfin	28779	Run Time	defbf534.h has incorrect PFDE_UART macro definitions

		Libraries	
Blackfin	28994	VDK	Potential for excessive stack usage on Blackfin processors
SHARC	28283	LDF	The section "seg_int_code" has grown unnecessarily
SHARC	28569	VDK	Sharc li2152 link warnings when using earlier Si Revision
SHARC	28692	Run Time Libraries	0.0 2126x libraries built with an inappropriate silicon revision
SHARC	28761	Run Time Libraries	No SRU header files
TigerSHARC	28263	Compiler	long long to double conversion fails in byte-addressing mode
TigerSHARC	28267	Compiler	Assertion failure: bril/zp/macdefs.c:2747 with -O -never-inline
TigerSHARC	28295	VDK	Cannot view system stack usage in the expert linker
TigerSHARC	28490	VDK	CCNTx register is read in the wrong order
TigerSHARC	28880	VDK	TS20x Idle thread prevents scheduling of user threads