ADS 1.2 Release Notes

January 2002

This document gives additional general information on the use of the ARM Developer Suite (ADS) release 1.2.

For an overview of the main differences between ADS 1.2, ADS 1.1, ADS 1.0.1 and earlier SDT (Software Development Toolkit), see the ADS 1.2 Getting Started Guide.

1 INSTALLATION

- Before you can use ADS 1.2, you will need to install a license file. Details of how to install the temporary license key provided with ADS 1.2 and how to request a permanent license file are contained in the ADS Installation and License Management Guide.
- ADS 1.2 uses a different license key from that used with ADS 1.1, ADS 1.0.1 or ADS 1.0. If you are upgrading from an earlier ADS, you will need to uninstall the earlier version, install ADS 1.2 and then request a new license file as detailed in the ADS Installation and License Management Guide.
- Once you have received your license file, if you require assistance with its installation, please consult our License Management FAQ within the Technical Support area of the ARM web site. Further advice on license installation can be obtained from license.support@arm.com. Ensure that you provide your ADS serial number and license file with all such enquiries.
- Technical support in the use of ADS 1.2 should be obtained from your supplier. Contact details should be provided on the sticker on the outside of your ADS 1.2 box and also within your license file. They will also be displayed when you enter '`armcc -help`' at the command line.
- The supported hardware platforms are described in the ADS Installation and License Management Guide. For Windows platforms, the following Versions/Service Packs are recommended:
  - Windows NT 4 - Service Pack 5 or above.
  - Windows 2000 - ServicePack 1
  - Windows 98 - Second Edition
  - Windows 95 - version 2

2 CODEWARRIOR

- **Projects from ADS 1.1 and earlier will be auto-converted to ADS1.2 format**

  ADS 1.2 uses CodeWarrior Pro7 IDE4.2.5 build 764 from Metrowerks. ADS 1.1 used CodeWarrior Pro5 IDE4.0.4. Metrowerks has changed the format of Access Paths and File Mappings target settings data between the two versions. ARM has also changed the ARM Debugger and ARM Runner settings data because ADW has become obsolete.
  When opening an existing ADS 1.1 (or earlier) project file with ADS 1.2, CodeWarrior will ask whether to convert to the new target settings format. The original project will be saved as project.old.mcp. Any projects using ADW as their debugger or runner will be changed to use AXD. If you used the ADW options '-armul' or '-adp', you may need to create and use AXD session files for ARMulator or remote_A respectively. If you used an ADW script file you will need to convert this manually to a script with the equivalent AXD commands.
- **Converted interworking projects may fail to build**
  ADS 1.1 interworking projects that are auto-converted by ADS 1.2 CodeWarrior may sometimes fail to build. To fix this:

  1) Check the current target's "Subtarget Output". There may be two files listed with the same name, the first being the real object file, the second being a rogue duplicate shown with a 'library' icon (with a horizontal

bar). Highlight the second rogue object, then use the "Project Inspector". If this file has no valid path, then delete it. The project should then rebuild OK.

2) Check the subtarget 'chaining', by expanding the Thumb target to reveal the ARM subtarget. The Chain column on the far right should show a black dot for the subtarget. If not, 'rechain' the subtarget by clicking carefully in the Chain column.

3) If CodeWarrior warns "File missing from Files view hierarchy. Some files were missing from the hierarchy information for the Files view of project yourproject.mcp. These files have been added to the Files view hierarchy in a group called "Rescued Items".", then rechain as above. Use the Project Inspector to check whether the rescued files have a valid path. If not, they may be deleted. [DE 68111]

- **ADS 1.1 CodeWarrior may crash when loading an ADS 1.2 project**
  When opening an ADS 1.2 project file with ADS 1.1, CodeWarrior may crash. To workaround this:

  1) Use 'File->Export Project...' to export the ADS 1.2 project as an XML file "project.mcp.xml."

  2) Start ADS1.1 CodeWarrior and use 'File->Import Project...' to import "project.mcp.xml"

  3) CodeWarrior will warn that some settings are newer and may be reset to Factory defaults. Press the Convert button to convert the project back to ADS1.1 format. [DE 81360]

- **Run/Debug buttons**
  A new 'Run' button should appear next to the 'Debug' button on the CodeWarrior toolbar. The Run button might fail to appear if ADS 1.2 is installed on a Win 98 or Win 2000 platform which previously had ADS 1.1 installed. To fix this, you must reset your project window toolbar to see the new icon. This can be done easily with a project window open, using View > Toolbars > Reset Window Toolbar. [DE 67166]

- **scf extension is used by "Windows Explorer Command"**
  CodeWarrior uses the *.scf file extension to denote a linker scatterloading file. Unfortunately, the *.scf extension is already mapped to the Windows Explorer command on WinNT, 98, 2000 and Me, which means that a scatterfile named, for example, 'scat.scf', is displayed as just 'scat' in a Windows Explorer view. On WinNT and 98, it is possible to remove the Windows Explorer Command as a registered file type (Explorer View | Options... | File Types tab). It is not possible to reassign the *.scf extension on Win2000 or Win Me as it does not appear in the registered file types extension list. However, it is possible to reassign this using the DOS command-line with, e.g:
  ```
  assoc .scf=TXTFile
  ```
  [DE 49058]

- **ARM/Thumb interworking projects on a network drive give Error : (Fatal) L6002U: Could not open file**
  This link failure may occur with ADS 1.2 CodeWarrior ARM/Thumb interworking projects which are located on a network drive. ADS 1.2 ARM/Thumb interworking projects will use a "Directory of objects" link-step for the ARM objects by default, instead of creating a partially-linked object (as is done in ADS 1.1 and earlier). Unfortunately, when the files are on a network drive, CodeWarrior" fails to convert "`\drivename\dir\subdir`" to "`\\drivename\dir\subdir`". The workaround is to map the network drive to a mapped drive, e.g. map "`\\drivename\dir`" to (say) "`f:`". After doing this, you should reopen the project from the mapped drive, remove object code and then re-build the project. [DE 90209]

- **Error : (Fatal) L6002U: Could not open file nozi**
  This link failure may occur with ADS 1.1 CodeWarrior projects which are converted by ADS 1.2 CodeWarrior, then rebuilt. This has been seen with certain AFS 1.3 projects. The simple fix is to remove the "-F nozi" from the command line, then rebuild. The same applies to "norw" and "noro". [DE 74011]

- **Makefile Importer Wizard does not offer ARM Linker as target**
  The CodeWarrior IDE Guide, section 3.3.5, "Importing makefiles into projects" describes how to use the MakeFile Importer Wizard. Unfortunately, the 'ARM Linker' does not appear in the drop-down list because of some missing lines in makefile.dat. Contact your supplier to obtain the corrected file, quoting [DE 87359]

- **comctl32 version**
  If Buttons in project view or text window do not work, or a Text window reports Unhandled Exceptions when closing CodeWarrior IDE, it is very likely that you have a version of comctl32 which is insufficiently mature because the Installer failed to install it or you have not rebooted after install. CodeWarrior IDE requires version 4.71 or later. To check the version and upgrade please run

  ```
  <ADSv1_2 install>\bin\50comupd.exe
  ```

You must do the upgrade as an Administrator on NT. You can also check the version by selecting comctl32.dll in your Windows System (winnt\system32) directory and looking at the properties version page (alt+Enter, context menu, right mouse Properties).

- **It is advisable to use small fonts**
  Small fonts allow some of the target settings panels to be used more easily. Use Start->Settings->Control panel->Display->Settings->Font Size to switch to small fonts. After that, to improve legibility use Start->Settings->Control panel->Display->Appearance->Scheme->Windows Standard (large). Large fonts will cause ARM tools to be displayed in a scrolling window, but buttons and field labels in CodeWarrior IDE panels have their text compressed or cut off. [DE 30583]
- **Implicit template instantiations** generate one definition per compilation unit.
  This might result in multiple definitions of a template item in the browser. [DE 32129]
- **Adding include paths**
  You should not add include files to CodeWarrior IDE by using the –I option in the 'Equivalent Command Line' dialog box of the compilers' Language Settings. If you do this, CodeWarrior IDE has no knowledge of the included files and you will lose Browse information and Error processing associated with the included files. Instead, you should use Target->Access Paths. [DE 28939]
- **Interleaved C/assembler output (armcc -S -fs) from CodeWarrior**
  The compilers support the "-S" and "-S -fs" options. -S generates an assembly source file (.s) from the C or C++ source. -S -fs generates a text file of C or C++ source code interleaved with the assembly code generated. You can add either of these switches to the "Equivalent Command line" in the target setting window (Alt+F7).

  "-S" will create a text file named "source.c.s" from a source file "source.c". However, because a .s file is created, CodeWarrior adds it to the project, and will then invoke armasm, and attempt to assemble it. This assemble step will succeed because the output from -S is a fully assemblable source file. "-S -fs" will create a text file named "source.c.txt" from a source file "source.c", and the link step will fail as expected. It is not possible to use the -o switch to explicitly name the output file to prevent this behaviour, because the equivalent command line must be suitable for use with all C sources in the project.

  You may wish to create a new build Target (e.g. "DebugRel_listings"), that uses the same build settings as your 'normal' build, but with the addition of "-S" or "-S -fs" to generate the necessary output file. [DE 43159]
- **Synchronization problem with files on remote server**
  CodeWarrior can sometimes lose synchronisation between files in your working directory and files on a remote server (which CodeWarrior caches).
  To fix this, change 'Target settings, Build extras', 'Use modification date caching' to OFF (for all targets). When ON, CodeWarrior caches the date/time stamps of files in the project, which helps to speed up the build process. However, this also means that CodeWarrior is not always able to determine if a source file has been modified outside the immediate CodeWarrior environment. This applies if you are using a third-party editor, or for multi-user development environments where source files can be modified and checked in through version control systems. For more details, refer to the CodeWarrior IDE Guide, section 9.3.3. [DE 72760].

3 CORE TOOLS

Linker:

- "Undefined symbol" error when scatterloading
  If you are using a scatter-file as input to the linker, you **must** provide a reimplemention of `__user_initial_stackheap()` to set the heap and stack boundaries. If you do not provide this function, the linker will report:
  `Error: L6218E: Undefined symbol Image$$ZI$$Limit (referred from sys_stackheap.o).`

  This can occur with ADS 1.2, even though the same project built OK with ADS 1.1, for the following reasons: When armlink is creating an image using a scatter-loading description, it creates some region-related symbols as described in Linker and Utilities Guide, section 4.1.1, "Region-related symbols". These symbols are created by armlink only if your code references them. In ADS 1.2, the symbols `Image$$RW$$Base, Image$$RW$$Limit, Image$$RO$$Base, Image$$RO$$Limit, Image$$ZI$$Base, and Image$$ZI$$Limit` are undefined when a scatter-loading description file is used. In ADS 1.1, these symbols took the value zero. The default implementation of `__user_initial_stackheap()`provided in the C libraries uses

`Image$$ZI$$Limit and SWI SYS_HEAPINFO (0x16)`. Therefore you must reimplement `__user_initial_stackheap()` and define a value for the start of the heap region and the top of the stack region. See the section on library memory models in the Compiler and Libraries Guide and the section on Writing code for ROM in the Developer Guide for more information. [DE 58562].

Example implementations of `__user_initial_stackheap()`for C and assembler are given below. For example projects which illustrate these, see ADS\Examples\embedded\embed\retarget.c.

1) Example implementation in C (for the default one region memory model)

```
#include <rt_misc.h>

__value_in_regs struct __initial_stackheap __user_initial_stackheap(
        unsigned R0, unsigned SP, unsigned R2, unsigned SL)
{
    struct __initial_stackheap config;
    config.heap_base = 0x60000; // select a safe address for the heap base
    config.stack_base = SP;     // inherit stack base from the environment
    return config;
}
```

2) An equivalent implementation of 1) in assembler:

```
    EXPORT __user_initial_stackheap
__user_initial_stackheap
    LDR   r0,=0x60000
    MOV   pc,lr
```

3). To place the heap directly above a ZI region (and inherit the stack base from the environment):

```
    IMPORT ||Image$$region_name$$ZI$$Limit||
    EXPORT __user_initial_stackheap
__user_initial_stackheap
    LDR r0, =||Image$$region_name$$ZI$$Limit||
    MOV pc, lr
```

where region_name is the name of an execution region as defined in your scatter description file.

4) Example implementation in C (for the two region model, and code built with –apcs /swst)

```
#include <rt_misc.h>
#pragma import(__use_two_region_memory)

__value_in_regs struct __initial_stackheap __user_initial_stackheap(
        unsigned R0, unsigned SP, unsigned R2, unsigned SL)
{
    struct __initial_stackheap config;
    config.heap_base = 0x60000;
    config.heap_limit = 0x70000;
    config.stack_base = 0x50000;
    config.stack_limit = 0x40000;
    return config;
}
```

C/C++ Compilers

- When compiling for stack checking, sp may be dropped below sl-256 by an amount not exceeding the size of the register save area. The solution is to allow more than 256 bytes of slop below sl:
  - 256+56 = 312 bytes if your program makes no use of floating-point hardware (all integer registers but sp and pc can be saved).
  - 256+56+96 = 408 bytes if your program uses FPA (all FPA registers can be saved).
  - 256+56+132 = 444 bytes if your program uses VFP (all VFP registers can be saved + 1 "map" word from FSTMX).

If you use the memory models supported by the ARM C library and you use stack stacking, you must claim the extra stack slop by implementing the function __user_stack_slop (see Compiler and Libraries Guide, section 4.9.7).

A suitable implementation in assembly language (build using armasm -apcs /swst) is:

```
        AREA ||.text||, CODE, READONLY
        EXPORT __user_stack_slop
user_stack_slop PROC
        MOV      r0,#56    ; or #56+96 or #56+132
        MOV      r1#0      ; or your value...
        MOV      pc,lr
        ENDP
        END
```

In C (build using armcc -apcs /swst), use:

```
    struct __stack_slop {int always, cleanup;};
    __value_in_regs struct __stack_slop __user_stack_slop(void)
    {
        struct __stack_slop slop;
        slop.always = 56;         /* or 56+96 or 56+132 */
        slop.cleanup = 0;         /* or your value */
        return slop;
    }
```

Be aware that __user_stack_slop only exists as a weak reference, so if __user_stack_slop is the only function in a section, the linker will remove it. To prevent this, use the -keep __user_stack_slop option when linking. [DE 41901]

- If you select a CPU which implies an FPU (such as "-cpu ARM10200", which implies VFP), but also select a different FPU (e.g. softvfp), the compiler will compile the code for whatever FPU is specified last. For example,

-cpu ARM10200	correctly gives hard VFP
-fpu softvfp -cpu ARM10200	gives hard VFP, silently
-cpu ARM10200 -fpu softvfp	compiles code as soft VFP with (correctly) hard VFP build attributes

If you use the command line then your FPU choice must come *after* your CPU choice. Similarly, if you use CodeWarrior you should set the FPU *after* you set the CPU.

Note that ARM1020T is an equivalent CPU but with no VFP (so no implied FPU setting), so if you want to compile for ARM10 but do not require hardware VFP then use '-cpu ARM1020T -fpu softvfp' instead of '-cpu ARM10200 -fpu softvfp'. [DE 37665]

- Overload resolution of conversion operators does not consider the implicit “this” argument. [DE 32449]

Profiling:

- To profile assembler language functions you must mark the functions with `FUNCTION` and `ENDFUNC` directives, for example:

```
   AREA test, CODE
   EXPORT asm_func
asm_func  FUNCTION

; your assembler code here
   mov pc, lr
   ENDFUNC

   END
```

See the Assembler Guide for details

- Callgraph profiling cannot be used on a program built ROPI (Read Only Position Independent) if it uses function pointers or it is launched at any address other than its preferred base address
- The profiler does not understand virtual function calls.
- Profiling assumes ip (r12) is unused over function calls, so take care when using assembler.

- The profiler will not recognise Thumb BLX <register> instructions correctly. You must compile Thumb code containing variable function calls as Architecture 4T, if you wish to use callgraph profiling.

Assembler:

- Very large assembler files under Windows
  Under Windows, assembler source files must be CR-LF terminated if the total source size exceeds the assembler's cache size (default 8MB). The most likely visible effect is that the assembler appears to be looping. Use the –maxcache option to change the assembler's cache size if necessary [DE 25955].

4 DEBUGGERS

- Debugger-to-target support in ADS 1.2 is:
  - AXD, armsd, ARMulator and Remote_A all support RDI 1.5.1, the latest version of the Remote Debug Interface.
  - AXD supports ARMulator, Remote_A, Multi-ICE 1.4, Multi-ICE 2.0 and Multi-ICE 2.1.
  - AXD additionally supports RealMonitor, via RMHost (RealMonitor.dll) through to the RMTarget.
  - armsd supports ARMulator and Remote_A, but not Multi-ICE.
  - EmbeddedICE is supported (via Remote_A) by AXD and armsd, but it must have ICEAgent version 2.07 EPROM fitted.
- To use AXD on Solaris, HP-UX or Linux, an X display that supports 24-bit color (TrueColor) is recommended. The typical Linux default 16-bit color display is **not** suitable. [DE 84609]
- AXD for Linux runs on Redhat Linux 6.2 and 7.1 only. It will **not** run on Redhat Linux 7.2, or on 7.1 if its glibc library has been updated from 2.2.2 to 2.2.4. [DE 97609]
- When using AXD's CLI commands under Linux, error messages are incorrectly truncated, e.g. setproc 3 returns "The index doesn't co" [DE 86310]
- When running AXD on Linux, if a segfault occurs in libX11.so.6 (locale), then try setting the locale to default with:
  (for csh and tcsh): `setenv LANG C`
  (for sh & ksh): `export LANG=C`
  [DE 84509]
- AXD does not display banked User/System registers correctly when running in FIQ mode. AXD incorrectly shows r8_usr - r12_usr as having the same values as r8_fiq - r12_fiq. If the mode is subsequently switched to User or System, all the registers display correctly. [DE 84559]
- If ADS 1.2 is installed, deinstalled, then installed again in a different directory, running AXD from the Start menu will report:
  `DBE Warning 00066: Could not change to directory 'C:\Program Files\ARM\ADSv1_2\Bin'.`
  where the directory is where ADS was _previously_ installed. To fix this, you should either configure a target (with Options->Configure Target), or locate AXD's default session file (named default-1-2-0-0.ses) and delete it. [DE 41937]
- AXD on Solaris can appear to 'hang' when focus is changed during in-place editing. For example, if you double-click on a register in the Registers window to edit it, then double-click on a different register. To fix this, you should modify the X-Windows configuration file. In the users $HOME there should be a file called .Xdefaults, if not it needs to be created. The file needs to contain the line
  `Dtwm*secondariesOnTop: True`
  Take care to enter this *exactly* as given – case is important! [DE 41213]
- If a hardware watchpoint is set on e.g. a global variable, code execution will stop whenever that address is written to, even if the value written is the same. You will see the message:
  `DBE Warning 00085: The target has stopped because of a hardware breakpoint or watchpoint`
  when the same value is written (i.e. the contents have not changed). Conditional hardware watchpoints are also broken, because code execution will always stop when the watchpoint is hit, even if the condition has not yet been met. Software watchpoints behave correctly, stopping only if the content actually changes. [DE 44055]

- Can't set hardware watchpoints on global/static objects above Stack Pointer [DE 42061]
  It is not possible to set a hardware watchpoint on e.g. a global variable which resides at an address above the stack pointer (SP). AXD wrongly assumes that the global is on the stack because it resides at an address above the stack pointer (SP). Watchpoints on stack variables always use software watchpoints, so a hardware watchpoint will not be used.
  The workaround is to specify the address of the global variable, instead of its name, when setting the watchpoint. The address of the variable can be found using e.g.: '`print &variable`'. Suppose this returns the address 0xA00000, a hardware watchpoint can then be placed on the variable (assuming the target supports this) with AXD's CLI: '`wpt 0xA00000`'
- When connecting to non-stop targets (e.g. RealMonitor) which are already running, AXD is unable to ascertain the current execution context. In this case the current pc is set to 0 and the visual indicators will be positioned according to this address. [DE 39119]
- When using Multi-ICE with AXD, watchpoints can be set on individual bytes within a word. However, a change to a watched byte is only detected if the byte itself is addressed and its value changed. A change to the location is not detected if it occurs due to an instruction changing a word or half-word that contains the byte, unless the address of the word or half-word is also that of the byte. This happens because Multi-ICE programs the registers of the watchpoint unit of the EmbeddedICE logic of the core, so that execution is stopped when any data write to the specified address occurs. Only exact address matches stop program execution. Suppose you want to watch the byte at address 0x3. This could be modified by:

  1) a write (of any size) to address 0x3, or
  2) a halfword write to address 0x2, or
  3) a word write to address 0x0.

  There is no way to program the EmbeddedICE logic to trap all 3 conditions, so Multi-ICE uses only 1). [DE 42613]
- AXD provides a multi-processor enabled interface, but there are no targets that can support this feature in this release. Multi-processor support for Multi-ICE is provided by running multiple copies of AXD.
- AXD uses scripting to support the Command Line Interface, (CLI). This is implemented using VBScript.DLL which is an OCX component. OCX components need to be "registered" before they can be utilized by other applications. The version of VBScript shipped with ADS is 5.0. There may be other versions on your machine. The AXD installer will copy the 5.0 version into the ADS bin directory, but will NOT overwrite existing versions in your system directories or elsewhere. AXD will not function correctly with versions of VBScript earlier than 5.0, specifically the command line interface will not be available. This means the version of VBScript registered must be at least 5.0. However applications which rely on versions earlier than 5.0 may not function with later versions. It is possible to have only one version of the OCX registered at any one time.

  If your system is currently relying on a pre-5.0 version being registered, and you wish to use the AXD command line interface, it is suggested that you temporarily register the version in the ADS bin directory. When not using AXD you should re-register your existing version to return to normal system usage. The bin directory contains the tool regcomif.exe which will register the supplied OCX. This program may be used to register either the supplied OCX or to re-register an earlier version if this is necessary. It is simply invoked with the name of the DLL as a parameter:
  `regcomif <file>`

  Note that the program uses the Win32 LoadLibrary call which may attempt to load the file from the directory containing the program itself. It is therefore safer to specify a full pathname if there is any uncertainty as to which one might be found. Some options are available (listed by the -h option) but are unlikely to be needed. Note that registering a component will silently remove any registration of an earlier version.
- When running AXD on HP/UX and connecting to Multi-ICE, the message "`Wind/U Error (240): Function SetCursor, GDI API's must only be called from the primary thread on this platform`" may appear in the console window. This is harmless and can be ignored. [DE 37903]
- Loading images into AXD from the command-line (with -debug and -exec) may fail on HP/UX systems which have auto-mount set up, when the image is loaded via an auto-mount point. To avoid this, specify the full path name, e.g. `axd -debug /tmp_mnt/home/mydir/myimage.axf`.
  This problem is caused by an HP/UX system bug, which can be fixed by applying these patches:
  ACE5 (Dec 99 or later) - These are Additional Core Enhancements
  QualityPack (Dec 99 or later) - The Quality Pack is a pack of cumulative patches that HP releases every 6 months. [DE 37165]

- How to execute a script file when launching AXD

  It is possible to execute a script file when launching AXD, to set some debug target properties, e.g. launch AXD from DOS with:
  ```
  axd -script go.txt
  ```

  where go.txt contains:

  ```
  load image.axf
  spp vector_catch 0
  spp semihosting_enabled 0
  swat $top_of_memory 0x400000
  etc...
  ```

  You can also launch AXD with a script file from CodeWarrior, by configuring the target settings. In the 'Equivalent Command Line', enter:
  ```
  axd -script go.txt
  ```

  then press 'Save'.

5 ARMULATOR

- The ARMulator Configuration on-line help in AXD contains information that is incorrect. The help text currently reads:
  "If your processor uses an MMU, and it is enabled, select DEFAULT_PAGETABLES. If your processor uses a PU, or an MMU that is disabled, select NO_PAGETABLES."
  but should read:
  "If your processor uses an MMU or PU, and you want it to be enabled, select DEFAULT_PAGETABLES. If you want them to be disabled, select NO_PAGETABLES." [DE 84563]
- The Debug Target Guide, section 4.14.3 describes how to change the cache or TCM size of an ARMulator model of a synthesizable processor like the ARM926EJ-S. Unfortunately, the example shown for the ARM926EJ-S is incorrect. The text should read:

  ```
  1. Open your copy of the processors.ami file.
  2. Insert a new section:
  {ARM926EJ-S=Processors_Common_ARMULATE
   IRamSize=0x80000
  }
  ```

  [DE 48603]
- ARMulator's configuration dialog box now allows adding a choice of VFP models to any processor. Where processors already contain one (e.g. ARM10200E, ARM10200T), you must select NO_FPU for proper operation. [DE 83611]
- To enable simulation of floating point hardware when using ARM9, ARM10 or XScale processors, select e.g. VFPv2 from the Floating Point Unit list. Do not check the "Floating Point Emulation" box because this (FPE) is incompatible. [DE 83209]
- On Unix, if you dismiss ARMulator's configuration dialog and AXD's "Configure Target" dialog, by pressing the return key too quickly, the "Configure Target" buttons may stop working. Wait a moment before pressing return, or use the mouse instead. [DE 82959]
- ARMulator produces a spurious debugger internal variable: $acmd. This should be ignored. [DE 86815]

6 REMOTE_A

- Users of EPI's JEENI with an ADS debugger via remote_a should ensure they have the latest version of the JEENI firmware fitted, currently version 2.2. Contact EPI for more details at: http://www.episupport.com/

7 ANGEL

- These releases of Angel are supported in ADS:
  - Angel 1.31, as released with ADS 1.0 and ADS 1.0.1
  - Angel 1.20, as released with SDT 2.50 and SDT 2.51
  - Angel 1.04, as released with SDT 2.11a

ADS does not support Angel 1.02, as released with SDT 2.11, or any earlier releases of Angel.

Angel is now supplied as part of the ARM Firmware Suite (AFS), not on the ADS CD.

8 TRACE

To use the ARM Trace Debug Tools with ADS 1.2, you **must** have TDT 1.1.1. ADS 1.2 will **not** work with TDT 1.1 or earlier. TDT 1.1 works with ADS 1.1. TDT 1.0 works with ADS 1.0/1.0.1 only.

9 DOCUMENTATION

- You cannot search for the following characters in DynaText searches. The first set is not described within the DynaText documentation, but the second set is described.

  _ $ - . (i.e. underscore, dollar, minus, period)

  : ; ? * ( ) = < > " [ ] (i.e. colon, semicolon, question mark, asterisk, open round bracket, close round bracket, equal, less than, greater than, double quote, open square bracket, close square bracket).

  A workaround is to use the PDF documents. If you did not install the PDF documents as part of your original installation, you can install them by invoking the ADS installation again, and choosing the PDFs.
- DynaText allows you to search for occurrences of words, phrases, and patterns. This means that you must use wildcard characters if you want to search for a pattern which may not be a complete word. For example you should specify 'option*' if you want to find all references to both 'option' and 'options' (and 'optional'). If you specify 'option', you will not find references to 'options' (or 'optional'). Read the DynaText Documentation for full details of how to use DynaText.
- Illegible text when viewing DynaText files under WinMe
  When viewing DynaText files using Windows Me, the Chapter titles may be illegible, because Dynatext's colour scheme conflicts with the default Windows Me Desktop colour scheme. This can be resolved by changing the colour scheme using Window's Display properties. [DE 50605]
- Dynatext searches will fail to find an occurrence of a word if part of that word has a different format. For example, Dynatext will not find an occurrence of the word "foobar" if it is formatted as "foobar" (with foo underlined). The workaround is to search for "foo*".

Revision History:

14th Dec 2001 - 1st Release

22nd Jan 2001 - Added DEs 97609, 90209, 48603, profiling with `FUNCTION` and `ENDFUNC` directives.