Atmel Studio 6.1

Welcome to Atmel Studio

Atmel Studio is the integrated development environment from Atmel. It provides you a modern and powerful environment for doing AVR® and ARM development.

Get started by exploring the included example projects. Run your solution on a starter or evaluation kit. Program and debug your project with the included simulator, or use one of the powerful on-chip debugging and programming tools from Atmel®. Get productive with the various navigate, refactor and intellisense features in the included editor. Experience seamless integration with various Atmel web services like Atmel Video Lounge[1], Atmel Store[2] and datasheets to keep you updated and help you to design your solutions.

With strong extension possibilities and online gallery[3], it is possible for both designers and 3rd party to provide plug-ins and customize the environment for best use and productivity.

Atmel Studio carries and integrates the GCC toolchain for both AVR® and ARM, Atmel Software Framework, AVR Assembler and Simulator. All newest Atmel tools are supported including Embedded Debugger, AVR ONE!, JTAGICE mkII, JTAGICE3, STK500, STK600, QT600, AVRISP mkII, AVR Dragon™ and SAM-ICE™.

For a detailed list of supported devices and tools please see "Supported Devices" on page 19.

---

[1] http://atmel.com/video/
[2] http://store.atmel.com
[3] http://gallery.atmel.com

Table of Contents

1. Installation Instructions

1.1 System requirements

1.1.1 Supported operating systems

- Windows XP (x86) with Service Pack 3 - all editions except Starter Edition

- Windows Vista (x86 & x64) with Service Pack 1 - all editions except Starter Edition

- Windows 7 (x86 and x64)

- Windows 8 (x86 and x64)

- Windows Server 2003 (x86 & x64) with Service Pack 2

- Windows Server 2003 R2 (x86 and x64)

- Windows Server 2008 (x86 and x64) with Service Pack 2

- Windows Server 2008 R2 (x64)

1.1.2 Supported architectures

- 32-Bit (x86)

- 64-Bit (x64)

1.1.3 Hardware requirements

- Computer that has a 1.6GHz or faster processor

- RAM

    - 1 GB RAM for x86

    - 2 GB RAM for x64

    - An additional 512 MB RAM if running in a Virtual Machine

- 4 GB of available hard disk space

- A minimum display resolution of 1024 x 768 or higher is recommended

1.2 Downloading and installing

- Download the latest Atmel Studio installer.

- Atmel Studio 6.1 can be run side by side with Atmel Studio 6.0, AVR Studio® 5.0 and 5.1. Uninstallation of AVR Studio 5.0 or 5.1 is not required.

- Verify the hardware and software requirements from "System Requirements" section.

- Make sure you are logged on with Administrative privileges.

- Please save all your work before starting, because the installation might prompt you for a restart if required.

- Please disconnect all Atmel USB/Serial hardware devices.

- Windows XP users are urged to stay connected to the internet during installation to allow for automatic certificate updates. See FAQ for more information.

- Double click the installer executable file. Please note that this might take some time to extract depending on H/W configuration.

- Atmel Studio Prerequisites installation will start. NOTE : If you have all the prerequisites already installed then this dialog will not be shown.

  - If .NET Framework 4.0 is not already installed, the installer will start the .NET Framework setup, Note that this does not include SP1.

  - Accept the license agreement and proceed through the installation. If the installer prompts for restart please do so. After restart the installation will start automatically.

- If Visual Studio Isolated shell 2010 is not installed, the installer will start the Microsoft Visual Studio Isolated Shell (2010) Setup.

  - Accept the license agreement and proceed through the installation.

- The Atmel USB Driver will install/upgrade the existing Jungo USB driver, and will also keep the existing hardware to work.

  - Accept the license agreement and proceed through the installation.

  - Note: If you have previous versions of Jungo USB driver then the installer will update them. The Atmel USB driver (Jungo USB Driver + SAM USB Driver) is fully compatible with its previous versions. So AVR Studio 4, Studio 32 and AVR Studio 5.0, 5.1 should continue to work with the updated driver without any issues.

- After this, Atmel Studio installation should start.

- Click "Next" to continue.

- Accept the license agreement and continue.

- Choose the "Destination Directory to Install" and click "Next".

- Review the summary and click "Install".

- The installation will copy all files and prompt to click "Finish".

- At the end, the installer will display options to associate files with AVR Studio. Please choose them if you prefer to open files with the mentioned file extensions in Atmel Studio.

- The installer also displays an option to "Start Atmel Studio after completion". If you choose to open, then please note that Atmel Studio will launch with administrative privileges, since the installer was either launched as administrator or with elevated privileges (in Windows Vista or later).

2. FAQ

1. **What is the Atmel USB Driver ?**

   The Atmel USB Driver is a cumulative installer that bundles the Jungo USB driver for the AVR tools and the Segger USB Driver for SAM tools.

2. **Will Atmel Studio 6.1 work in parallel with Atmel Studio 6.0, AVR Studio 5.0, 5.1, AVR Studio 4.0 and AVR32 Studio?**

   Yes, it will work. Note: When you are uninstalling AVR Studio 4.0 or AVR32 Studio please be careful when you manually delete folders or registry entries after uninstall, as there might be other keys and folders deployed by Atmel Studio inside Atmel folder and registry paths.

3. **I have AVR Studio 4 in my PC. When installing Atmel Studio it updated the Jungo USB driver. Will AVR Studio 4 still work?**

   Yes, it will work. If Jungo driver is already present and its version is anything less than the new one, then the installer will update the Jungo driver you already have. The updated Jungo driver is fully compatible with its previous versions. So AVR Studio 4 and Studio 32 should continue to work with the updated driver without any issues.

4. **During installation the installer hangs more than 20 minutes while displaying: "Atmel Studio requires the following items to be installed on your computer." A progress bar is running but there is no indication about real progress**

   If you have network connectivity that is limited (no internet connectivity), please disconnect the network or disable all the active network adapters and start installation of Atmel Studio. This happens only on Windows XP and Windows 2003 based machines.

5. **The Atmel Studio installer crashes when installing using the "run as" option in Windows XP (any architecture) with "Protect my computer ..." option enabled.**

   Don't use the "run as" option with the "Protect my computer ..." option enabled. The installer will have only read-only access to some of the registry hives and system folders, causing it to malfunction. This is due to the DEP option enabled by windows in this mode. Refer to http://blogs.msdn.com/b/aaron_margosis/archive/2004/09/10/227727.aspx for more information.

6. **Atmel Studio and Visual Studio 2010 SP1 conflicts**

   If Atmel Studio is installed on a machine with Visual Studio 2010 SP1, the following message might be displayed when starting up Atmel Studio or Visual Studio: "Only some of the Microsoft Visual Studio 2010 products on this computer have been upgraded to Service Pack 1. None will work correctly until all have been upgraded".

   Follow the instructions in the error message dialog to resolve the problem.

7. **Atmel Studio cannot find any debuggers or programmers after uninstalling AVR Studio 4.**

   The AVR Studio 4 installer uninstalls the Jungo USB Driver which is shared by Atmel Studio. Repairing the Atmel Studio installation might not reinstall Jungo USB Driver. So Download the Atmel USB Driver installer from the Atmel web and install it.

8. **Atmel Studio cannot find any debuggers or programmers when Norton AntiVirus is running**

   Atmel Studio might not show any connected tools if Norton AntiVirus is running. To make it work make sure Norton AntiVirus allows avrdbg.exe to communicate with the Atmel Studio front-end by adding avrdbg.exe as an exception in the Norton AntiVirus allowed programs. This is the same with any anti-virus program that by default blocks ports.

9. **Windows shows a message box with the following message when attempting to run Atmel Studio installer: " Windows cannot access the specified device, path or file. You may not have the appropriate permissions to access the item. "**

   This might be caused by an anti-virus program blocking the installation of the Atmel Studio. We have seen this with the Sophos antivirus package. Temporarily disable the Sophos service running on the machine (or any corresponding anti-virus service), and attempt installation.

10. **After installing and starting Atmel Studio in Windows XP, the following error is displayed. "This application has failed to start because MSVCR100.dll was not found. Re-installing the application may fix this problem."**

This probably means that your PC doesn't have Service Pack 3 for Windows XP installed. Please install SP3 and then install Atmel Studio again.

11. **Atmel Studio takes a very long time to start on my PC, but runs well in a VM environment**

    The Visual Studio shell (and thus Atmel Studio) does a considerable amount of processing during start-up. Parts of the operations are WPF operations which benefits greatly by updated graphics libraries and drivers. Installing the latest graphics driver may give a performance boost both during normal operation and during start-up.

12. **Verification and programming often fails a with a serial port buffer overrun error message when using STK500**

    This is a known issue. Due to DPC latency, serial communication can have buffer overruns on the UART chipset. E.g. other drivers , thus causing A workaround which works for most systems is to use an USB to serial adapter.

13. **Internet Explorer 6 does not display user documentation correctly**

    This is a known issue, primarily due to the incompatibility of the version of HTML used to prepare the help content and the IE6 rendering engine. Please update your browser to a later version of IE to resolve the issue. This problem is only observed with Windows XP and Windows Server 2003.

14. **Help Registration fails with error code 401.**

    One of the common reasons for this failure is because of an corrupted Help Store location. The help store location is usually %SYSTEMDRIVE%\ProgramData\Microsoft\HelpLibrary for Vista or later machines and it is %SystemDrive%\Documents and Settings\All Users\Application Data for Windows XP and Windows Server 2003 based machines. Please clean up this location and try running the installer again.

15. **When launching from a guest account, the following error is displayed when starting Atmel Studio: "Exception has been thrown by the target of an invocation"**

    Atmel Studio neither installs under guest account and nor runs under it.

16. **Can we install and run Atmel Studio from within a Virtual Machine**

    Yes, with simulator there should be no issues. However with physical devices like debuggers and programmers, the VM must offer support for physical USB and Serial port connections.

17. **How can I reduce the startup time of Atmel Studio?**

    ● Make sure you have uninstalled unwanted extensions in studio.

    ● Disable "Allow Add-in components to load".

        1. Go to Tools → Options → Add-in/Macro Security.

        2. Then uncheck the Allow Add-in components to load option.

    ● Disable the startup page (alternative 1)

        1. Disable Show page on Startup in start page.

    ● Disable the startup page (alternative 2)

        1. Go to Tools → Options → Environment → Startup → At Startup.

        2. Select Show empty environment option.

18. **How to improve studio performance on XP OS or Single Core System?**

    ● Make sure your system has the latest version of the Windows Automation API.

    ● Disable Visual Experience, if you have single core system:

        1. Go to Tools → Options → Environment → General.

        2. Then uncheck the Automatically adjust visual experience based on client performance, Enable rich client visual experience and Use H/W graphics acceleration options.

- Disable the Editor Track Changes

    1. Go to Tools → Options → Text Editor.

    2. Then uncheck the Track Changes option.

- Disable the Track Active Item in Solution Explorer

    1. Go to Tools → Options → Projects and Solutions → General.

    2. Then uncheck the Track Active Item in Solution Explorer option.

19. **How to improve studio performance on any OS?**

- Make sure your system has the latest version of the Windows Automation API.

- Exclude the following directories and files from **antivirus scanner**:
    - **Atmel Studio Installed directory** and its sub directories and files
    - **%AppData%\Roaming\Atmel directory** and its sub directories, files
    - **%AppData%\Local\Atmel directory** and its sub directories, files
    - Your **Atmel project directories**

- Visual Studio 10 shell requires a lot of swap space. Increase the paging file. Also put the system to maximize performance. Both options are found in System → Properties → Performance → Settings menu.

20. **Should I install the latest Windows Automation API 3.0?**

    Yes, if your OS is any of the following

- Windows XP
- Vista
- Server 2003
- Server 2008

21. **How can I make sure my system has the latest Windows Automation API 3.0?**

    Your system has the latest Windows Automation API if you have Windows 7 or Windows 8 operating system. Only Windows XP, Windows Vista, Windows Server 2003 and Windows Server 2008 have the old version API. Find the `UIAutomationCore.dll` file in your system (normally found in the windows folder) and compare the version number of that file. The **FileVersion** will be 7.X.X.X. for the new API. Latest API can be found at http://support.microsoft.com/kb/971513.

22. **My Project is large and it takes a long time to open. Is there any option to avoid this delay?**

    Visual Assist X parses all the files when we opening the existing project. You could disable this option:

    1. Go to VAssistX → Visual Assist X Options → Performance.

    2. Then uncheck the Parse all files when opening the project.

23. **I have the limited RAM size in my system and I work long hours in the same instance of Atmel Studio. After some time, Atmel Studio becomes slow on my system. What is the reason for this?**

    Press **Ctrl**+**Shift**+**Alt**+**F12** *twice* to force Atmel Studio to garbage collect.

24. **Does Atmel Studio perform better on multi-core processors than on single-core?**

    Yes, Atmel Studio performs better on a multi-core system.

25. **How can I make my projects build faster ?**

    You can enable parallel build Option from Tools → Options → Builder → GNU Make → Make Parallel Execution Of Build. This option will enable GNU make utility's parallel execution feature. The build output window log messages may not be in proper order.

3. New and Noteworthy

3.1 Atmel Studio 6.1

- Support for Embedded Debugger platform

- Support for Xplained Pro kits

- Atmel Software Framework 3.8.1

- AVR 8 Toolchain 3.4.2 (with upstream GCC 4.7.2)

- AVR 32 Toolchain 3.4.2 (with upstream GCC 4.4.7)

- GCC ARM Embedded Toolchain 4.7.3. Note! The default toolchain for ARM is now based on the branch from the GCC ARM Embedded project.

- CMSIS 3.20

- Updated Visual Assist

- Command line utility for firmware upgrade

- Stimulus for simulator. Create a stimuli file to write register values while executing simulation.

- Support for new devices

3.2 Atmel Studio 6.0

- Support for Atmel ARM-based MCUs with Atmel SAM-ICE

- Atmel Software Framework 3.1.3

- AVR Toolchain 3.4.0

- ARM Toolchain 3.3.1

- Atmel Software Framework Explorer

- Support for QTouch Composer as extension

- Updated Visual Assist

- New extension gallery

3.3 AVR Studio 5.1

- New version of AVR Software Framework (ASF)

- Availability and installation of new ASF versions through extension manager, without having to upgrade Studio 5

- Support for side by side versioning of ASF, with the ability to upgrade projects

- Syntax highlighting and better debugging support for C++ projects

- Support for importing AVR 32 Studio C++ projects

- New version of AVR Toolchain

- New command line utility (atprogram) with support for all Atmel AVR tools and devices

- Enhancements to programming dialog including support for ELF programming

- New version of Visual Assist with several enhancements and bugfixes

- Support for 55 new devices

4. Atmel Studio Extensions

Extensions and updates to Atmel Studio are available through the Atmel Gallery. Access it through http://gallery.atmel.com or use the extension manager in Atmel Studio under the 'Tools' menu.

The set of extensions supporting a given Atmel Studio version may vary. Visit http://gallery.atmel.com to see which extensions are available for a given version of Atmel Studio. Note that at the time of the release of Atmel Studio 6.1 not all extensions have been ported from 6.0.

Atmel Studio 6.1 does not automatically reinstall extensions installed on Atmel Studio 6.0.

5. Bugs and New Features

5.1 New features

Issue #AVRSV-3231:

Unable to add additional makefiles to the master makefile.

Issue #AVRSV-3517:

Atmel Studio editor was sluggish when editing very large files (>15k lines).

Issue #AVRSV-3647:

#pragma region and #pragma endregion was not supported as CodeFolding pair.

Issue #AVRSV-3654:

#pragma message "Show this message" was not shown as Information in Error List.

Issue #AVRSV-3656:

Switches -funsinged-char and -funsigned-bitfields are missing in default toolchain settings.

Issue #AVRSV-3707:

Debugging state is not preserved when object file is re-loaded.

Issue #AVRSV-3917:

DEBUG and NDEBUG are added as preprocessor definitions for debug and release configurations respectively.

Issue #AVRSV-4092:

AVR Libc documentation is now included in the built-in help system for F1 lookups, search and indexing.

Issue #AVRSV-4140:

User can report issue to Atmel Studio team through the menu "Help -> Report a Bug".

Issue #AVRSV-4188:

Switches -Wextra and -fdata-sections are added to Toolchain settings UI.

Issue #AVRSV-4541:

Entry File for Avr Assembler projects was not easily visibly identifiable.

5.2 Notable bugs fixed

Issue #AVRSV-197:

"break when hit count is equal to" and "break when hit count is greater than or equal to" doesn't work in Atmel Studio. Both these alternatives were behaving as "break when hit count is multiple of".

Issue #AVRSV-1023:

Incorrect number of GPIO ports may be displayed for UC3 devices in I/O view.

Issue #AVRSV-2061:

Visual Assist extension does not show conditionally preprocessed values.

Install Naggy extension to have this feature.

Issue #AVRSV-2546:

Some mega devices have mapped IO displayed as read-only in the memory view when debugging using JTAG. The memory can be written using the I/O View or the immediate window.

Issue #AVRSV-2708:

Some structs are displayed incorrectly in the watch view.

Issue #AVRSV-2862:

Build fails when the command line to compiler/linker exceeds 8k characters.

Issue #AVRSV-2875:

Data type 'long double' is not properly supported in watch view.

Issue #AVRSV-2902:

To view the content of the variable use "BY " , "WO " , or "DW " before the location in the watch view. This will treat it as a pointer that can be dereferenced to byte, word, or dword respectively.

E.g. instead of watching "VariableName", watch "BY VariableName".

Issue #AVRSV-3152:

Programming internal flash doesn't update the progressbar.

Issue #AVRSV-3161:

J-LINK Ultra and Pro tools are not supported.

Issue #AVRSV-3252:

Atprogram and daisy chain: The order of the daisy chain arguments are reversed.

Issue #AVRSV-3315:

Programs does not start executing after using "start without debugging" with TPI on AVRISP mkII.

Issue #AVRSV-3450:

Step Out (Shift F11) may not work correctly for SAM3 and SAM4 devices if the call stack has more than two frames.

Issue #AVRSV-3482:

atprogram does not recognize sam devices unless first part of name is written in uppercase.

Issue #AVRSV-3494:

Invalid data memory usage displayed during the build process.

Issue #AVRSV-3511:

Multiple instances of atprogram might interfere with each other and make tools temporarily unavailable.

Issue #AVRSV-3537:

When using AVR Dragon, the programming dialogue in Atmel Studio 6 doesn't show ATmega8 and ATmega8A in the drop down list of supported devices.

Issue #AVRSV-3544:

Automatic enabling of the debugWire interface fails on devices missing the CKDIV8 fuse.

Issue #AVRSV-3545:

Programming UC3 is slow compared to AVR Studio 5.1.

Issue #AVRSV-3571:

Extensions (vsix) fail to install by the main installer program.

Issue #AVRSV-3572:

Stepping on sei(); does not set I flag.

Issue #AVRSV-3573:

Changes in included header files in ASM files in C project are not reflected till cleanup.

Issue #AVRSV-3575:

ATSAM3X8H Flash Size incorrectly reported during programming.

Issue #AVRSV-3588:

Stop debugging when running hangs backend for UC3 targets.

Issue #AVRSV-3606:

When "Use tool polling" is enabled, an active Segger session may be closed unexpectedly.

Issue #AVRSV-3614:

Links in AVR assembler project should be relative.

Issue #AVRSV-3615:

Debugging of ATmega32HVB revision B does not work.

Issue #AVRSV-3621:

When a project directory contains a lot of files and folders, and 'Show All Files' command is invoked, it takes a lot of time to display the items.

Issue #AVRSV-3626:

Each launch session would leak a number of objects due to a missing destructor call. In the case of large ELF files with e.g. lots of macros (compiled with -g3), this could exhaust the backend memory in a few launches resulting in allocation failure and backend crashes.

Issue #AVRSV-3632:

Wrong criterion used for selecting clock source caused a slowdown in programming speed by a factor of 10 for some devices.

Issue #AVRSV-3648:

Importing AVR Studio 4 projects with .C (upper case) files fails to compile.

Issue #AVRSV-3664:

Debugger jumps to address 0 of flash if 'Reset' command is issued in disassembly view, even if BOOTRST is enabled.

Issue #AVRSV-3683:

Status register in processor view on ARM does not show correct data.

Issue #AVRSV-3794:

Unsupported instructions in assembler only creates "Warning" and not "Error" messages.

Issue #AVRSV-3836:

Changes of interface settings in programming dialogue is sometimes not applied when using SAM-ICE.

Issue #AVRSV-3908:

Unlocking SAM4L devices is not possible after setting the security bit.

Issue #AVRSV-3966:

Unable to watch some objects in C++ projects.

Issue #AVRSV-3972:

atprogram does not perform real chip reset for SAM3 & SAM4 devices before running flash code.

Issue #AVRSV-3983:

Some SAM user board projects may fail at link time.

Issue #AVRSV-3984:

atprogram error when trying to set ISP frequency with the interface parameter "-i ISP" in uppercase.

Issue #AVRSV-4015:

ATSAM3U4E GCC C project fails to program.

Issue #AVRSV-4040:

Problem with missing spaces in makefile options of imported AS4 project.

Issue #AVRSV-4048:

Output file generated by external makefile shall equal the name of the project with appropriate extension (say, projectname.elf) in order for debugging to succeed.

Issue #AVRSV-4055:

Unable to install Atmel Spaces extension.

Issue #AVRSV-4084:

Stack overflow will cause callstack evaluation to run indefinitely.

Issue #AVRSV-4117:

When programming certain hex files to an Xmega device, the programming window throws an error "[ERROR] Verifying Flash...Failed! address=0x1800 expected=0x24 actual=0xff".

Issue #AVRSV-4220:

atprogram doesn't perform chiperase correctly for ATSAM4L devices.

Issue #AVRSV-4222:

Programming of dual flash on ATSAM3U is not supported.

Issue #AVRSV-4422:

Uninstalling Atmel Studio 6.0 after installing Atmel Studio 6.1 beta will break the 6.1 installation. To fix the 6.1 installation after an uninstall of 6.0 run the 6.1 installer again and choose 'Repair'.

Issue #AVRSV-4442:

$(ToolchainDir) is not properly saved to project when added to additional include paths of compiiler settings.

Issue #AVRSV-4470:

Code folding in some cases works only when comments are added after opening and closing braces.

Issue #AVRSV-4473:

Projects imported from 6.0 to 6.1 get wrong debugging properties. Resolve issue by open up Solution explorer->project->properties, re-select the debugger options and Save.

Issue #AVRSV-4474:

Interrupt flag is not restored on AVR8 when single stepping and InterruptFree debugging is enabled.

Issue #AVRSV-4514:

If the PC has more than one listed COM port, studio is unable to add any STK500 to the tool list.

Issue #AVRSV-4544:

RFR2 devices are not listed at Device Programming dialog.

Issue #AVRSV-4578:

Firmware upgrade tool should clean up temporary files.

Issue #AVRSV-4580:

The ADC instruction is not visible in the Assembler documentation.

5.3   Known issues

Issue #AVRSV-283:

A message saying "webproperties.tlb could not be located" can be displayed on some systems. A workaround for this problem is to make a copy of a file named "webproperties???.tlb" in "C:\Program Files (x86)\Common Files\microsoft shared\MSEnv" (on the same location) and rename it to "webproperties.tlb".

Issue #AVRSV-389:

Conditional and Hit Count breakpoints works for the simulator, but not for hardware debuggers like the JTAGICE3, AVRONE etc.

Issue #AVRSV-414:

Power toggle and external reset is not handled gracefully in all situations.

Issue #AVRSV-546:

The .NET Framework installer might not work properly if network connectivity is limited. If connectivity is limited please disconnect from the network or disable all active network adapters before starting installation of Atmel Studio.

Issue #AVRSV-628:

Scrolling memory view does not work properly.

It is not possible to use the slider in the memory view to scroll it. Only the up and down arrows works.

Issue #AVRSV-680:

Sometimes breakpoints that are set in the Source Editor are not reflected correctly in the Disassembly Window while debugging.

Issue #AVRSV-769:

The following ASF modules are not available as standalone in the "Select Drivers from the ASF" menu, but only as examples: XMEGA Sleep Manager and ADC driver, AVR UC3 USB Stack from ASF v1, ECC Hamming, TLV320AIC23B codec,FAT file system with play list support, Joystick interface (5-way), MEMORY - EBI SDRAM Controller, MEMORY - MCI - MultiMedia Card Interface, MEMORY - SD/MMC card access using MCI, MEMORY - SD/MMC card access using SPI, MEMORY - NAND Flash on EBI, MEMORY - AT45DBX DataFlash, TOUCH - AT42QT1060 QTouch 6-channel sensor, MEMS Sensors - Accelerometer LIS3L06AL, TIMING - CS2200 Clock Synthesizer, LodePNG, FreeRTOS minimal, lwIP, Micrium uC/OSII, H&D Wi-Fi SPB Firmware Download.

Issue #AVRSV-831:

Atmel Studio installation may fail on XP systems if the Microsoft Windows Imaging Component (32-bit) is not installed. Please install this component, downloadable from http://www.microsoft.com/download/.

Issue #AVRSV-966:

Running the Atmel Studio installer using the "run as" option on Windows XP may crash the installer.

Issue #AVRSV-1192:

Internet Explorer 6 will not render the navigation menu in the user documentation correctly.

Issue #AVRSV-1254:

The asf.h header file is not included in all examples.

Workaround: Include this file manually if you add additional drivers using the

"Select Drivers from AVR Software Framework" dialog.

Issue #AVRSV-1527:

Using Chinese letters in file path in programming dialog is not fully supported.

Issue #AVRSV-1533:

Visual Studio 2010 RC/Beta version has conflict with RTM version of Microsoft Visual Studio 2010 Isolated Shell. The workaround is to uninstall Microsoft Visual Studio 2010 isolated Shell that is installed with Atmel Studio.

Issue #AVRSV-1557:

Mapped network drives do not appear in Project Location window when creating a new project.

Issue #AVRSV-1603:

When debugging on ATmega16[A] or ATmega32[A] devices it is not possible to read out the value of UBRRH using the debugger.

Issue #AVRSV-1675:

If a driver for a tool has not been installed (first time it's plugged in) and the user plugs the tool into the PC when Atmel Studio is running then the front-end will show the tool in the "Available Tools"-view but the back-end will not have access to the tool as a OS driver for the tool does not exist. Any operation on the tool initiated from the front-end will fail as it's not available to the back-end.

Restart Atmel Studio to access the tool.

Issue #AVRSV-1733:

Stepping in source view over a software reset may leave the target running on Xmega devices.

Issue #AVRSV-1758:

Projects which include paths with non-Latin characters are not supported.

Issue #AVRSV-1760:

Service Pack 1 of Visual Studio 2010 installed on a PC where Atmel Studio 6 is installed, may initiate a need for reapplying the SP1 installer. A dialog box will then appear during startup of Atmel Studio, and detail the steps that must be taken.

Issue #AVRSV-1883:

The IO window does not fully support registers like e.g. DIRSET, DIRTGL and DIRCLR for the XMEGA family (used to manipulate a corresponding DIR register). Toggling the value of bits in these registers have undefined result on DIR.

Issue #AVRSV-1888:

Debugging ATmega128 in ATmega103 compatibility mode is not supported.

Issue #AVRSV-1895:

'Alt + G' does not open the file <avr/io.h>. This file is not parsed by Visual Assist.

Issue #AVRSV-1901:

Creating two projects in the same solution which have different devices is not supported. Create two different solutions instead.

Issue #AVRSV-2022:

Running Folding@home together with Atmel Studio may cause unresponsive user interface. We recommend to disable the Folding@home when running Atmel Studio.

Issue #AVRSV-2163:

Files/Folders with more than one consequent spaces are not supported as part of AVRStudio 5 projects.

Issue #AVRSV-2558:

HVPP for ATtiny2313A does not work on STK500.

Issue #AVRSV-2601:

During installation of Atmel studio, visual studio 2010 shell installation will re-register the vsjitdebugger. This might make VS2008 and VS2005 versions of visual studio not to get a chance to debug a crashed application reported by windows.

Workaround : Run repair with visual studio on top of atmel studio installation. This should re-enable the capability of visual studio to get a chance to handle crashed applications.

Issue #AVRSV-2884:

Project creation may fail when file/folder name of the project or its sub-items name exceeds 256 characters limit.

Issue #AVRSV-3261:

GCC C project - Build fails with " '.stack' will not fit in region 'ram' " for SAM devices with RAM less than 8k.

This error is due to that default stack size is set to 0x2000 (8k). For devices where size for the RAM section is set to 8k in the linker script, there will not be room for other data in the RAM section and this error will happend.

The linker script says that the symbol STACK_SIZE should be defined by the user. To define this symbol , add '--defsym=STACK_SIZE=0x1000' (or use size value that is relevant for your project) to the 'ARM/GNU Linker -> Miscellaneous -> Other options' under Toolchain properties of your project.

Issue #AVRSV-3272:

If installing AVR Studio 6.0, 5.1 or 5.0 after installing Atmel Studio 6.1, the installer of Jungo usb-driver generates a warning. Please ignore this and let the installation continue.

Issue #AVRSV-3312:

Printing help from Microsoft Help Viewer may print a subset of pages. Workaround is to open the page URL in another web browser and print from that. URL can be found by pressing F4 in the help viewer.

Issue #AVRSV-3313:

If you encounter the error : variable 'xxxx' must be const in order to be put into read-only section by means of '_attribute_((progmem))', then this description applies. The problem is due to the incompatibility of the ASF source code with the AVR GCC compiler. The GCC 4.6 Release document (http://gcc.gnu.org/gcc-4.6/changes.html) mentions that the error is expected and to use the ASF projects created in 5.1 (i.e ASF 2.9.0) we have to use avr gcc toolchain verison 3.3.1 and for later ASF versions use 3.4.0. Alternatively you could manually add the const qualifier to the variable(s) that are reported, when compiling ASF 2.9.0 projects with AVR GCC toolchain 3.4.0 or later.

Issue #AVRSV-3376:

If a custom destination folder is selected in the Atmel Studio installer, some components will still install themselves in the default location.

Issue #AVRSV-3672:

ASF projects cannot be created in UNC paths. To create the ASF project, map the UNC path and assign a drive letter to the UNC path.

Issue #AVRSV-3695:

There is a limitation to the GCC support for AVR 8 bit devices with larger than 128k flash. Known workarounds exist for many of the problems this issue arises. Please consult The toolchain documentation regarding the -mrelax & -mshort-calls to learn more about this.

Issue #AVRSV-3761:

When atprogram.exe is run & interrupted in between a programming sequence; for the next invocation to the same programming tool it is throwing an error.

A hardware reset of the tool will resolve the issue.

Issue #AVRSV-3993:

On virtual machines like virtualbox, the event endpoint does not work properly and no events will be propagated from the tool to our backend.

Issue #AVRSV-4037:

Using a SEGGER tool, after a chiperase in the programming dialogue, reading device flash to file does not reflect the true, (erased) state of the device.

Issue #AVRSV-4050:

User signature on RFR parts can only be accessed by JTAG or parallel programming.

Issue #AVRSV-4166:

Changing SAM devices on the "device"-tab for an ASF project does not update the device everywhere in the project. The following must be manually updated: i) Device Name in ARM/GNU Common, ii) ARM/GNU C Compiler > Symbols, iii) ARM/GNU Linker > Misc > Linker FLags T../src/asf/sam/utils/linker_scripts/<update device name>/gcc/flash.ld, iv) ARM/GNU Preprocessing Assembler > General > Assembler Flags.

Issue #AVRSV-4216:

After installing an extension, some windows, notebly the Memory Windows, can pop up floating on the screen and leaving a blank memory window docked. If this happens, please rearrange the windows as desired. This should fix the issue.

Issue #AVRSV-4337:

Build abruptly fails in 6.0 Atmel studio without proper error message and the error window shows no error.

Tail of the Build Output:

Task "RunCompilerTask"

========== Build: 0 succeeded or up-to-date, 1 failed, 0 skipped ==========

Reason: Project file was upgraded from 6.0 to 6.1.

Steps to Restore back the project to working condition:

Scenario 1: (With Backup)

Check whether there is a back up project in the projectfolder with the name ProjectName_6_0 (For Example the backup project is GccApplication1_6_0.cproj if the actual project name is GccApplication1.cproj)

* Project with the name GccApplication1.cproj is the upgraded project to confirm edit the project file in editor you should be able to see <ProjectVersion>6.1</ProjectVersion>.

* Open the project GccApplication1_6_0.cproj in Atmel Studio 6.0. It should prompt you to save the solution file. Save and build it should work fine.

Scenario 2: (Without Backup)

If the backup project is not found in the project folder chances are that you would have upgraded the project from 6.0 to 6.1 without opting for the backup.

* Edit the project file modify <ProjectVersion> tag and set the version to 6.0 and also modify the <ToolchainName> tag by removing .C or .CPP from the tag (For example com.Atmel.AVRGCC32.C must be renamed as com.Atmel.AVRGCC32) build the project now.

Issue #AVRSV-4380:

When building a project in Atmel Studio, and if you get an error like the one as follows

<some file>.o: No such file or directory

during the linking stage, then it could be because of the number of characters in the command line. Windows expects that the command line be less than 8192 characters.

To workaround the issue, reduce the name of the folder so that the command line becomes shorter.

Issue #AVRSV-4431:

Selecting Erase Chip after programming fails for SAM4L, even if it claims OK. Workaround is to select Erase Chip a second time.

Issue #AVRSV-4440:

The SAM header files have been updated, due to this there are breaking changes when upgrading from 6.0 to 6.1. Bare bone SAM projects created with Atmel Studio 6.0 can get compilation errors due to changes in defines. You can continue to use the old headers by keeping Atmel Studio 6.0 and 6.1 installed in parallel and use the toolchain from 6.0. ASF projects are not affected.

Issue #AVRSV-4501:

Toolchain libraries "Full Path" property will display the base path of the current toolchain.

Issue #AVRSV-4521:

Double click the "Libraries" node to know the status of Libraries physical presence.

Issue #AVRSV-4576:

Modifying EEPROM data values in the memory view during debugging of XMEGAE5 devices causes the EEPROM data to be corrupted.

6. Supported Devices

The following tables lists all supported tools and devices.

6.1 AVR UC3

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
AT32UC3A0128	D/P		D/P	D/P	D/P			D/P		P
AT32UC3A0256	D/P		D/P	D/P	D/P			D/P		P
AT32UC3A0512	D/P		D/P	D/P	D/P			D/P		P
AT32UC3A1128	D/P		D/P	D/P	D/P			D/P		P
AT32UC3A1256	D/P		D/P	D/P	D/P			D/P		P
AT32UC3A1512	D/P		D/P	D/P	D/P			D/P		P
AT32UC3A3128	D/P		D/P	D/P	D/P					P
AT32UC3A3128S	D/P		D/P	D/P	D/P					P
AT32UC3A3256	D/P		D/P	D/P	D/P					P
AT32UC3A3256S	D/P		D/P	D/P	D/P					P
AT32UC3A364	D/P		D/P	D/P	D/P					P
AT32UC3A364S	D/P		D/P	D/P	D/P					P
AT32UC3A4128	D/P		D/P	D/P	D/P					P
AT32UC3A4128S	D/P		D/P	D/P	D/P					P
AT32UC3A4256	D/P		D/P	D/P	D/P					P
AT32UC3A4256S	D/P		D/P	D/P	D/P					P
AT32UC3A464	D/P		D/P	D/P	D/P					P
AT32UC3A464S	D/P		D/P	D/P	D/P					P
AT32UC3B0128	D/P		D/P	D/P	D/P					P
AT32UC3B0256	D/P		D/P	D/P	D/P					P
AT32UC3B0512	D/P		D/P	D/P	D/P					P
AT32UC3B064	D/P		D/P	D/P	D/P					P
AT32UC3B1128	D/P		D/P	D/P	D/P					P
AT32UC3B1256	D/P		D/P	D/P	D/P					P
AT32UC3B1512	D/P		D/P	D/P	D/P					P
AT32UC3B164	D/P		D/P	D/P	D/P					P
AT32UC3C0128C	D/P		D/P	D/P	D/P					P
AT32UC3C0256C	D/P		D/P	D/P	D/P					P
AT32UC3C0512C	D/P		D/P	D/P	D/P					P
AT32UC3C064C	D/P		D/P	D/P	D/P					P
AT32UC3C1128C	D/P		D/P	D/P	D/P					P
AT32UC3C1256C	D/P		D/P	D/P	D/P					P
AT32UC3C1512C	D/P		D/P	D/P	D/P					P
AT32UC3C164C	D/P		D/P	D/P	D/P					P
AT32UC3C2128C	D/P		D/P	D/P	D/P					P
AT32UC3C2256C	D/P		D/P	D/P	D/P					P
AT32UC3C2512C	D/P		D/P	D/P	D/P					P

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
AT32UC3C264C	D/P		D/P	D/P	D/P					P
AT32UC3L0128	D/P		D/P	D/P	D/P					P
AT32UC3L016	D/P		D/P	D/P	D/P			D/P		P
AT32UC3L0256	D/P		D/P	D/P	D/P					P
AT32UC3L032	D/P		D/P	D/P	D/P			D/P		P
AT32UC3L064	D/P		D/P	D/P	D/P	P		D/P		P
ATUC128D3	D/P		D/P	D/P	D/P					P
ATUC128D4	D/P		D/P	D/P	D/P					P
ATUC128L3U	D/P		D/P	D/P	D/P					P
ATUC128L4U	D/P		D/P	D/P	D/P					P
ATUC256L3U	D/P		D/P	D/P	D/P					P
ATUC256L4U	D/P		D/P	D/P	D/P					P
ATUC64D3	D/P		D/P	D/P	D/P					P
ATUC64D4	D/P		D/P	D/P	D/P					P
ATUC64L3U	D/P		D/P	D/P	D/P					P
ATUC64L4U	D/P		D/P	D/P	D/P					P

D: Debug is supported using one or more interface

P: Programming is supported using one or more interface

6.2   AVR Mega

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
AT90CAN128	D/P	P	D/P	D/P	D/P					P
AT90CAN32	D/P	P	D/P	D/P	D/P					P
AT90CAN64	D/P	P	D/P	D/P	D/P					P
AT90PWM1	D/P	P	D/P	D/P	D/P					P
AT90PWM161	D/P	P	D/P	D/P	D/P					P
AT90PWM216	D/P	P	D/P	D/P	D/P					P
AT90PWM2B	D/P	P	D/P	D/P	D/P					P
AT90PWM316	D/P	P	D/P	D/P	D/P					P
AT90PWM3B	D/P	P	D/P	D/P	D/P					P
AT90PWM81	D/P	P	D/P	D/P	D/P					P
AT90USB1286	D/P	P	D/P	D/P	D/P					P
AT90USB1287	D/P	P	D/P	D/P	D/P					P
AT90USB162	D/P	P	D/P	D/P	D/P					P
AT90USB646	D/P	P	D/P	D/P	D/P					P
AT90USB647	D/P	P	D/P	D/P	D/P					P
AT90USB82	D/P	P	D/P	D/P	D/P					P
ATA5790	D/P	P	D/P	D/P	D/P					P

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATA5790N	D/P	P	D/P	D/P	D/P					P
ATA5795	D/P	P	D/P	D/P	D/P					P
ATA5831	D/P	P	D/P	D/P	D/P					P
ATA5832	D/P	P	D/P	D/P	D/P					P
ATA5833	D/P	P	D/P	D/P	D/P					P
ATA6285	D/P	P	D/P	D/P	D/P					P
ATA6286	D/P	P	D/P	D/P	D/P					P
ATmega128	D/P	P	D/P	D/P	D/P			D/P	P[1]	P
ATmega1280	D/P	P	D/P	D/P	D/P			D/P		P
ATmega1281	D/P	P	D/P	D/P	D/P			D/P		P
ATmega1284	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega1284P	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega1284RFR2	D/P	P	D/P	D/P	D/P					P
ATmega128A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega128RFA1	D/P	P	D/P	D/P	D/P					P
ATmega128RFR2	D/P	P[2]	D/P	D/P[2]	D/P					P[2]
ATmega16	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega162	D/P	P	D/P	D/P	D/P				P	P
ATmega164A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega164P	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega164PA	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega165A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega165P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega165PA	D/P	P	D/P	D/P	D/P			D/P		P
ATmega168	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega168A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega168P	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega168PA	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega169A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega169P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega169PA	D/P	P	D/P	D/P	D/P			D/P		P
ATmega16A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega16HVA	D/P	P	D/P	D/P	D/P					P
ATmega16HVB	D/P	P	D/P	D/P	D/P			D/P		P
ATmega16HVBrevB	D/P	P	D/P	D/P	D/P					P
ATmega16M1	D/P	P	D/P	D/P	D/P					P
ATmega16U2	D/P	P	D/P	D/P	D/P					P
ATmega16U4	D/P	P	D/P	D/P	D/P					P
ATmega2560	D/P	P	D/P	D/P	D/P			D/P		P
ATmega2561	D/P	P	D/P	D/P	D/P			D/P		P
ATmega2564RFR2	D/P	P	D/P	D/P	D/P					P

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATmega256RFR2	D/P	P^2	D/P	D/P^2	D/P					P^2
ATmega32	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega324A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega324P	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega324PA	D/P	P	D/P	D/P	D/P	P		D/P	P	P
ATmega325	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3250	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3250A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3250P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3250PA	D/P	P	D/P	D/P	D/P			D/P		P
ATmega325A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega325P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega325PA	D/P	P	D/P	D/P	D/P			D/P		P
ATmega328	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega328P	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega329	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3290	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3290A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3290P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega3290PA	D/P	P	D/P	D/P	D/P			D/P		P
ATmega329A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega329P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega329PA	D/P	P	D/P	D/P	D/P			D/P		P
ATmega32A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega32C1	D/P	P	D/P	D/P	D/P					P
ATmega32HVB	D/P	P	D/P	D/P	D/P			D/P		P
ATmega32HVBrevB	D/P	P	D/P	D/P	D/P					P
ATmega32M1	D/P	P	D/P	D/P^2	D/P					P^2
ATmega32U2	D/P	P	D/P	D/P	D/P					P
ATmega32U4	D/P	P	D/P	D/P	D/P					P
ATmega48	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega48A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega48P	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega48PA	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega64	D/P	P	D/P	D/P	D/P			D/P		P
ATmega640	D/P	P	D/P	D/P	D/P			D/P		P
ATmega644	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega644A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega644P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega644PA	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega644RFR2	D/P	P	D/P	D/P	D/P					P

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATmega645	D/P	P	D/P	D/P	D/P			D/P		P
ATmega6450	D/P	P	D/P	D/P	D/P			D/P		P
ATmega6450A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega6450P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega645A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega645P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega649	D/P	P	D/P	D/P	D/P			D/P		P
ATmega6490	D/P	P	D/P	D/P	D/P			D/P		P
ATmega6490A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega6490P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega649A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega649P	D/P	P	D/P	D/P	D/P			D/P		P
ATmega64A	D/P	P	D/P	D/P	D/P			D/P		P
ATmega64C1	D/P	P	D/P	D/P	D/P					P
ATmega64HVE2	D/P	P	D/P	D/P	D/P					P
ATmega64M1	D/P	P	D/P	D/P	D/P					P
ATmega64RFR2	D/P	P^2	D/P	D/P^2	D/P					P^2
ATmega8	P	P	P	P	P			D/P	P	P
ATmega8515	P	P	P	P	P				P	P
ATmega8535	P	P	P	P	P					P
ATmega88	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega88A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega88P	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega88PA	D/P	P	D/P	D/P	D/P			D/P	P	P
ATmega8A	P	P	P	P	P			D/P	P	P
ATmega8HVA	D/P	P	D/P	D/P	D/P					P
ATmega8U2	D/P	P	D/P	D/P	D/P					P

Notes: [1] The ATmega128 needs the STK501 extension board to connect to the STK500.

[2] The user signature pages can only be accessed by JTAG programming or parallel programming, not by ISP programming.

D: Debug is supported using one or more interface

P: Programming is supported using one or more interface

6.3   SAM

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATSAM3A4C							D/P			
ATSAM3A8C							D/P			
ATSAM3N00A							D/P			
ATSAM3N00B							D/P			

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATSAM3N0A							D/P			
ATSAM3N0B							D/P			
ATSAM3N0C							D/P			
ATSAM3N1A							D/P			
ATSAM3N1B							D/P			
ATSAM3N1C							D/P			
ATSAM3N2A							D/P			
ATSAM3N2B							D/P			
ATSAM3N2C							D/P			
ATSAM3N4A							D/P			
ATSAM3N4B							D/P			
ATSAM3N4C							D/P			
ATSAM3S1A							D/P			
ATSAM3S1B							D/P			
ATSAM3S1C							D/P			
ATSAM3S2A							D/P			
ATSAM3S2B							D/P			
ATSAM3S2C							D/P			
ATSAM3S4A							D/P			
ATSAM3S4B							D/P			
ATSAM3S4C							D/P			
ATSAM3S8B							D/P			
ATSAM3S8C							D/P			
ATSAM3SD8B							D/P			
ATSAM3SD8C							D/P			
ATSAM3U1C							D/P			
ATSAM3U1E							D/P			
ATSAM3U2C							D/P			
ATSAM3U2E							D/P			
ATSAM3U4C							D/P			
ATSAM3U4E							D/P			
ATSAM3X4C							D/P			
ATSAM3X4E							D/P			
ATSAM3X8C							D/P			
ATSAM3X8E							D/P			
ATSAM3X8H							D/P			
ATSAM4E16E							D/P			
ATSAM4E8E							D/P			
ATSAM4LC2A							D/P			
ATSAM4LC2B							D/P			
ATSAM4LC2C							D/P			

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATSAM4LC4A							D/P			
ATSAM4LC4B							D/P			
ATSAM4LC4C							D/P			
ATSAM4LS2A							D/P			
ATSAM4LS2B							D/P			
ATSAM4LS2C							D/P			
ATSAM4LS4A							D/P			
ATSAM4LS4B							D/P			
ATSAM4LS4C							D/P			
ATSAM4S16B							D/P			
ATSAM4S16C							D/P			
ATSAM4S8B							D/P			
ATSAM4S8C							D/P			
ATSAM4SD16B							D/P			
ATSAM4SD16C							D/P			
ATSAM4SD32B							D/P			
ATSAM4SD32C							D/P			
ATSAM4SP32A							D/P			

D: Debug is supported using one or more interface

P: Programming is supported using one or more interface

6.4   AVR Tiny

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATtiny10		P						D/P		P
ATtiny13	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny13A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny1634	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny167	D/P	P	D/P	D/P	D/P					P
ATtiny20		P						D/P		P
ATtiny2313	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny2313A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny24	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny24A	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny25	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny26	P	P	P	P	P			D/P	P	P
ATtiny261	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny261A	D/P	P	D/P	D/P	D/P			D/P	P	P

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATtiny28									P	P
ATtiny4		P						D/P		P
ATtiny40		P						D/P		P
ATtiny4313	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny43U	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny44	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny44A	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny45	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny461	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny461A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny48	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny5		P						D/P		P
ATtiny828	D/P	P	D/P	D/P	D/P	P		D/P		P
ATtiny84	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny84A	D/P	P	D/P	D/P	D/P			D/P		P
ATtiny85	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny861	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny861A	D/P	P	D/P	D/P	D/P			D/P	P	P
ATtiny87	D/P	P	D/P	D/P	D/P					P
ATtiny88	D/P	P	D/P	D/P	D/P	P		D/P	P	P
ATtiny9		P						D/P		P

D: Debug is supported using one or more interface

P: Programming is supported using one or more interface

6.5   AVR Xmega

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATxmega128A1	D/P	P	D/P	D/P	D/P	P		D/P		P
ATxmega128A1U	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega128A3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega128A3U	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega128A4U	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega128B1	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega128B3	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega128C3								D/P		
ATxmega128D3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega128D4	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega16A4	D/P	P	D/P	D/P	D/P			D/P		P

Device	AVR Dragon	AVRISP mkII	AVR ONE!	JTAGICE3	JTAGICE mkII	QT600	SAM-ICE	Simulator	STK500	STK600
ATxmega16A4U	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega16C4	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega16D4	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega16E5	D/P	P	D/P	D/P	D/P					P
ATxmega192A3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega192A3U	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega192C3	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega192D3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega256A3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega256A3B	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega256A3BU	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega256A3U	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega256C3	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega256D3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega32A4	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega32A4U	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega32C4	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega32D4	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega32E5	D/P	P	D/P	D/P	D/P					P
ATxmega384C3	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega384D3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega64A1	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega64A1U	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega64A3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega64A3U	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega64A4U	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega64B1	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega64B3	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega64C3	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega64D3	D/P[1]	P	D/P	D/P	D/P			D/P		P
ATxmega64D4	D/P	P	D/P	D/P	D/P			D/P		P
ATxmega8E5	D/P	P	D/P	D/P	D/P					P

Notes: [1] XMEGA PDI mode on AVR Dragon does NOT work for the following XMEGA devices: A3/D3 - revisions B, C and E or A1 (up to revision K).

D: Debug is supported using one or more interface

P: Programming is supported using one or more interface

7. Device Notes

The following mature devices are not recommended for new designs:

- ATtiny11
- ATtiny12
- ATtiny15
- ATtiny22
- AT90S1200
- AT90S2313
- AT90S2323
- AT90S2343
- AT90S4433
- AT90S8515
- AT90S8535
- ATmega323
- ATmega161
- ATmega163
- ATmega103
- ATmega165
- ATmega169
- ATmega64HVE
- ATmega32U6
- AT90PWM2
- AT90PWM3
- AT90SCR100
- AT86RF401

See www.atmel.com for replacements.

8. Contact Information and Disclaimer

For support on Atmel Studio please contact **avr@atmel.com**.

8.1 Disclaimer

Atmel Studio is distributed free of charge for the purpose of developing applications for Atmel ARM and Atmel AVR processors. Use for other purposes is not permitted; see the software license agreement for details. Atmel Studio comes without any warranty.

9. Document Revision History

Doc. Rev.	Date	Comment
A	04/2013	First release

Atmel Enabling Unlimited Possibilities®