Freescale Semiconductor, Inc.
Release Notes

Kinetis Design Studio 3.2.0 Release Notes

1 Overview

The Kinetis Design Studio IDE is a complimentary integrated development environment for Kinetis MCUs that enables robust editing, compiling and debugging of your designs. Based on free, open-source software including Eclipse, GNU Compiler Collection (GCC), GNU Debugger (GDB), and others, the Kinetis Design Studio IDE offers designers a simple development tool with no code-size limitations. Furthermore, Processor Expert software enables your design with its knowledge base and helps create powerful applications with a few mouse clicks.

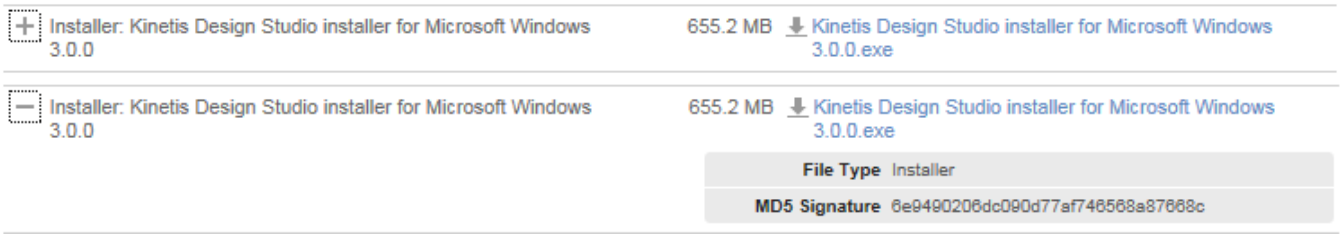
Contents

2 Installing Kinetis Design Studio

2.1 Before Installation

To verify downloaded installer you can compare MD5 signature of the downloaded product with checksum provided on the product download page.

1. On the KDS download page, click on the "+" sign in the left column, the MD5 Signature will show up:



2. Generate MD5 checksum of the downloaded installer:
   a. On Windows, you can use FCIV utility provided by Microsoft (https://support.microsoft.com/en-us/kb/841290):

      ```
      fciv -md5 ".\Kinetis Design Studio installer for Microsoft Windows 3.2.0.exe"
      ```

   b. On Linux, you can use build-in md5sum command:

      ```
      md5sum ".\Kinetis Design Studio installer for Linux 64-bit DEB 3.2.0.deb"
      ```

      or

      ```
      md5sum ".\Kinetis Design Studio installer for Linux 64-bit RPM 3.2.0.rpm"
      ```

   c. On Mac OS X, you can use build-in md5 command:

      ```
      md5 ".\Kinetis Design Studio installer for Mac 3.2.0.pkg"
      ```

3. Compare the two MD5 signatures

2.2 Installing on Windows

The Kinetis Design Studio software development tools are installed on Windows using the Windows Installer.

To install Kinetis Design Studio using the Windows installer:

1. Double-click the installer.

2. The Windows Installer initiates.
3. Click Next.
4. Follow the on-screen instructions and proceed through the installation.

2.3 Installing on Linux

To install the Kinetis Design Studio software development tools on a Linux system, use the following package files:

- .rpm — Use .rpm to install KDS software tools on systems using the RPM package manager. For example, Red Hat and CentOS.
- .deb — Use .deb to install KDS software tools on systems that use the Debian package manager. For example, Ubuntu.

2.3.1 Installing with Red Hat package manager (RPM)

To install the Kinetis Design Studio software development tools on a Linux Standard Base (LSB)-compliant system, use the .rpm package file:

```
$ sudo rpm -Uvh kinetis-design-studio-3.2.0-1.x86_64.rpm
Preparing ... ########################################### [100%]
1: Kinetis Design Studio ########################################### [100%]
```

This will install the Kinetis Design Studio software development tools to the default location (/opt/Freescale/KDS_v3).

2.3.2 Installing with Debian package manager (DEB)

To install the Kinetis Design Studio software development tools on Debian-like systems, including Ubuntu, use the .deb package file:

```
$ sudo dpkg -i kinetis-design-studio_3.2.0-1_amd64.deb
(Reading database ... .... files and directories currently installed .)
Preparing to replace kinetis-design-studio 3.2.0 (using kinetis-
designstudio_3.2.0-1_amd64 .deb) ...
Unpacking replacement kinetis-design-studio ...
Setting up kinetis-design-studio (3.2.0) ...
```

This installs the Kinetis Design Studio software development tools to the default location (/opt/Freescale/KDS_v3).

NOTE

KDS includes the GCC ARM Embedded toolchain, which is built for 32 bit hosts. If you are using a 64 bit system, be sure you have the appropriate 32 bit packages installed:

- For Ubuntu 1404 these packages are required to be installed: libc6:i386, libncurses5:i386, & libstdc++6:i386.

- For RPM based packages these packages are required to be installed: glibc.i686 and libncurses.so.5.

2.4 Installing on Macintosh

The Kinetis Design Studio software development tools are installed on MAC OSX using the MAC PKG installer. To install Kinetis Design Studio using the MAC installer:

1. Double-click on the installer (PKG) file.
2. The installer initiates
3. Click Continue.
4. Follow the on screen instructions and proceed through the installation.

NOTE

Currently only the Segger debugger works on MAC OSX. If you are using Freedom boards, ensure that you have the Segger OpenOCD firmware installed on the board which can be found on http://www.freescale.com/freedom and look for your particular freedom board's getting started page.

3 What's New in this Version

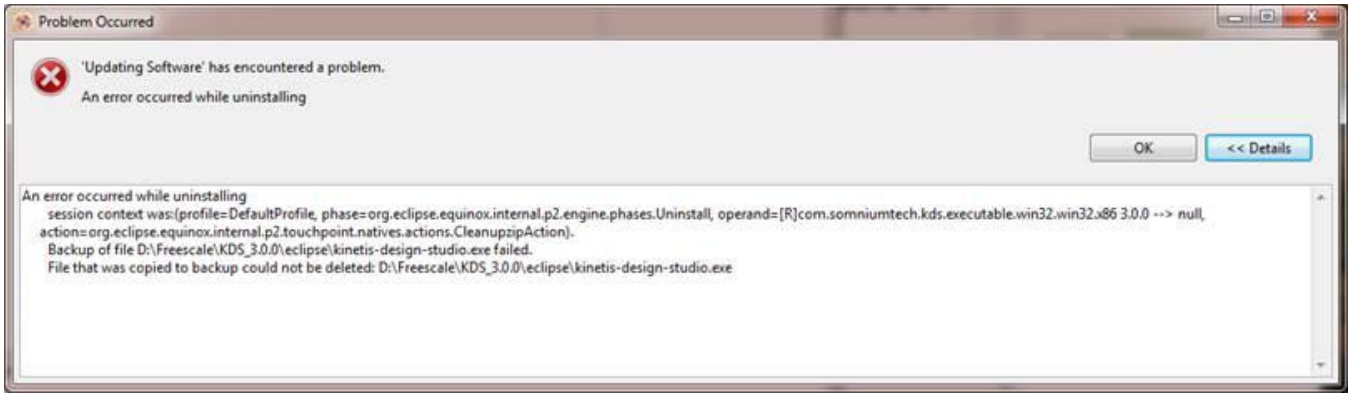
Kinetis Design Studio 3.2.0 update contains Processor Expert for Kinetis 3.0.2 (please refer to separate release notes) and additionally fixes the following problem:

- KDS-335: This version removes the features listed below, because they are blocking proper update of ARM Eclipse plugins from update site http://gnuarmeclipse.sourceforge.net/updates. The following features are removed:
  - Project-less debugging is not supported anymore - user is now required to have a project to be able to debug
  - Toolchain folder preference is now set to "GNU Tools for ARM Embedded Processors" installation instead of the previous "${eclipse_home}/../toolchain/bin" setting
- KDS-343: This version fixes startup problems when running on MAC OS X 10.11 ("El Capitan")
- KDS-178: New GNU ARM Build Tools for Windows removes 8192 characters limitation of commands to compiler and linker
- This version includes following updates:
  - Updated GDB ARM P&E Plugin v2.3.6 and Windows and Linux drivers v12.3
  - Updated Segger J-Link drivers 5.10n
  - Updated version of Project of Projects
  - New Kinetis SDK 2.x Project Wizard
  - Bare board project support for KM1x 50MHz and KM3x 50MHz/75MHz devices rev. A

4 Known Issues and Workarounds

For the latest information, training material and Frequently Asked Questions, visit the Kinetis Design Studio Community at https://community.freescale.com/community/kinetis-design-studio

- Administrative rights to Eclipse installation folder for updates: Users must have write access to the KDS installation directory to install new Eclipse plugins. This means that on Linux users need to launch KDS with root privileges when installing new plugins.
- KDS update from online site (using Help > Check for Updates) in some situations fail on Windows OS with following error:



This is caused by a known Eclipse issue:

https://bugs.eclipse.org/bugs/show_bug.cgi?id=427148

https://bugs.eclipse.org/bugs/show_bug.cgi?id=441098

Workaround: Start the KDS using KDS_v3\eclipse\eclipsec.exe, then make the update using Help > Check for Updates. After update is finished close the KDS and start it usual way.

- Conditional watchpoints and breakpoints: Conditional breakpoints and watchpoints, including those using ignore counts, do not work always.

  Workaround: do not use conditions for breakpoints and watchpoints, instead check for condition in the code and set a normal breakpoint.

- Symbolic Link to libudev for Linux: Like many other Linux packages, users of Ubuntu 14.04 must create a symbolic link to libudev.

  Workaround:

  ln -s /lib/x86_64-linux-gnu/libudev.so.1.3.5 /usr/lib/libudev.so.0

- Installation time on Ubuntu: Users attempting to install KDS using the Ubuntu Software Center may find that the Software Center claims to be installing for a long period of time, then returns to the start screen without emitting an error or installing the product. This is because the Software Center runs a quality checking tool, lintian, on the package before installing it. This tool is not implemented in a scalable manner, and doesn't handle the large KDS packages well. Users of high-end machines may find they are able to install successfully.

  Workaround: install using the command-line tool: dpkg

- Build binary not found: Occasionally after a successful build Eclipse does not find the built binary. This can manifest in a number of ways: The project does not show the Binaries meta-folder in the Project Explorer view, when the debug button on the debug toolbar is pressed the Debug configuration fails to launch a debug session because it reports binary file not found.

  Workaround: refresh the project folder (F5 under Windows).

- PEXMCU-531: Compiler error if using the TSS Processor Expert component. The compiler is because the component sources are using asm() instead of __asm().

  Workaround: Described in https://community.freescale.com/message/435546#435546

- KDS-223: Doing a reset command on the FRDM-K22F and OpenOCD debug connection gets stuck in the watchdog reset handler.

  Workaround: Use an alternative debug connection (P&E Multilink or SEGGER J-Link).
- KDS-190: Stepping over an endless loop (branch instruction pointing to itself) might fail with OpenOCD.

  Workaround: set a breakpoint on the loop statement.
- KDS-189: Debugging, downloading and stepping with OpenOCD is slow compared to other debug solutions.

  Workaround: Use alternative debug connections like P&E Multilink or SEGGER J-Link.
- KDS-240: The OpenOCD in KDS v3.0.0 is the same as in KDS v2.0.0, therefore any newer Kinetis devices are not supported with OpenOCD.

  Workaround: Use an alternative debug connection (P&E Multilink or SEGGER J-Link).
- Using TSS component: adding the TSS component for non-Kinetis SDK Processor Expert project will cause a compilation error.

  Workaround: Documented here: https://community.freescale.com/thread/330174
- Mac OS X and Eclox: The installation of the Doxygen Eclox Eclipse plugin (http://home.gna.org/eclox/) fails under Mac OX with an error message. That plugin is not maintained anymore and a fix is not likely.
- Moving/Using projects between different host operating systems: If using a project created/used on a different operating system (e.g. migrating a project from Windows to Linux), it is recommended to delete the output (usually named 'Debug') folder of the project to enforce proper regeneration of the make files, as a 'clean' operation alone might not be enough.

5 Revision History

Revision	Change description
Rev 1.0	Initial version

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
www.freescale.com/support