RN00294

Release Notes for S32 Design Studio 3.6.12 RFP

Rev. 2.0 — 26 October 2026 **Release notes**

1 Getting Started

NXP Semiconductors is pleased to announce the release of the S32 Design Studio for S32 Platform 3.6.12 for NXP Arm based devices and hardware accelerators. S32 Design Studio for S32 Platform is based on the Eclipse open development platform and integrates the Eclipse IDE, GNU Compiler Collection (GCC), GNU Debugger (GDB), and other software to offer designers a straightforward development tool with no code-size limitations.

This package contains:

- GNU Bare-Metal Targeted Tools for Arm 32-bit Embedded Processors:
  - GCC version 11.4 20230528, build 1763 revision gf703eb2
- GNU BareMetal Targeted Tools for Arm 64-bit Embedded Processors:
  - GCC version 11.4 20230528, build 1763 revision gf703eb2
- GNU Linux Targeted Tools for Arm 64-bit Embedded Processors:
  - GCC version 11.4 20230528, build 1763 revision gf703eb2
- Libraries: NewLib, NewLib Nano[1]
- Semihosting for Arm 32-bit and 64-bit bare-metal target toolchains.
- MSYS2 2022.06-1.
- Eclipse 2023-12 framework, CDT 11.4, and Java17.
- GDB 15.1 with Python 3.10.x support.
- S32 Debugger (with S32 Debug Probe) support for Arm cores.
- S32 Debugger provides OS Awareness support for FreeRTOS, OSEK, and Zephyr OS.
- S32 Trace for Arm Cortex-A53, R52, M33, M7 cores with S32 Debugger.
- S32 Flash Tool.
- PEmicro debugger support.
- Lauterbach Trace32 support.
- IAR debugger support.
- TASKING debugger support.
- Segger debugger support.
- Green Hills compiler support.
- IAR compiler support.
- Diab compiler 7.x support.
- The S32DS Extensions and Updates Tool.
- The SDK integration is provided with more software packages. The SDK packages can be installed and updated with the S32DS Extensions and Updates Tool.
- SDK Management.
- The migration support is provided to upgrade:
  - SDK version attached to the project,
  - GCC-based toolchain for project.

1 The availability of libraries depends on the toolchain version and the core type. Tools for Arm 64-bit processors support NewLib libraries only. Tools for Arm 32-bit processors do not support EWL libraries for the Cortex-R52 and Cortex-M33 cores.

- S32 Configuration Tools with important additions in Pins, Clocks, Peripherals, eFUSE, IVT, DCD, QuadSPI, DDR, FFC, GTM and SERDES tools.
- The wizards for creating application, library projects and projects from project examples for the supported processor families[2]
- The Getting Started page.
- Peripheral Registers view.
- Arm System Registers view.
- Watch registers view.
- Memory Spaces view.
- JTAG Daisy-Chain view.
- S32DS MMU Viewer.
- S32DS Debug Perspective.
- S32DS C/C++ Perspective.
- Threads (Zephyr RTOS) view.
- Smart card support for secure debugging.
- Multiple Elf Loading for S32 Debugger.
- Remote Probe Connection.
- Global Variables view.
- Debug Card view.
- OS Details browser view.
- Cross-Trigger support:
  - UI option to configure various states(source/destination)
  - Cross Trigger state control - enable, disable, clear

1.1 Installation

The S32 Design Studio for S32 Platform 3.6.12 RFP installation package contains the base tools and the installer. Support for the NXP Arm® based processor families is provided with more software packages. The packages can be installed on the product from the S32DS Extensions and Updates tool or downloaded from the product webpage.

Note: *The users are allowed to install S32 Design Studio for S32 Platform 3.6.12 RFP with non-admin rights, if components needing admin privileges (e.g., drivers or Visual C++ redistributables) are selected, the installer prompts once for elevation to install the components.*

Note: *The plug-ins to support third-party compilers or debuggers such as Lauterbach Trace32® are not included in the installation package and have to be installed from the corresponding sites or installation packages.*

Run the installation package. The wizard guides you through the installation process.

New functionality can be added to S32 Design Studio for S32 Platform 3.6.12 RFP with more software packages, updates, and patches. Software packages add support for specific NXP Arm® based processor families. Updates and patches extend the general functionality of the product and correct software defects.

New functionality can be added directly from the Internet or from a downloaded archive. If your computer is connected to the Internet, select **S32DS Extensions and Updates** on the **Help** menu to find and install all updates and packages available. If your computer does not have access to the Internet, you can download software packages, updates, and patches from the product page and install them from the archive files using the S32DS Extensions and Updates tool.

2 Support for wizards and the project examples are provided in the device-specific software packages.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

2 Release Specifics

2.1 Release Content

The release has RFP quality in terms of testing and quality documentation. Software Tools are delivered as RFP-level reference code, tested on engineering or qualification-intent samples, but they are **not certified or approved for vehicle production** and **must not be used in car production**. Any production-related certifications or approvals are the Customer's responsibility.

2.2 Supported Features

- Integrated updated versions of the tools packages:
  - S32 Debugger Core 3.6.12
  - S32 Trace 3.6.12
  - S32 Configuration Tool 1.9.7
  - S32 Flash Tool 2.4.4
- The S32 Design Studio 3.6.12 installer includes the newest Platform and Platform Tools package 3.6.12, GCC 11.4, public development packages for S32N5 version 3.6.12, S32R47 version 3.6.9, for S32K3 version 3.6.7, for SAF85 and S32K1 version 3.6.6, for S32G and SAF86 version 3.6.5 and for S32M2, S32Z2/E2, S32R41 and S32R45 version 3.6.4.

2.3 Recommended configuration

Windows

- PC with a 2.6 GHz Intel Pentium compatible processor or better.
- 8 GB of RAM.
- 30 GB of disk space (when installing all product features or all updates).
- 24 GB of temporary storage (required only during the product installation).
- USB port for communications with target hardware.
- Ethernet port for communications with target hardware (optional).

Linux

- PC with a 2.6 GHz Intel Pentium compatible processor or better.
- 8 GB of RAM.
- 25 GB of disk space for installation files (full product or updates).
- 20 GB of temporary storage (required only during the product installation).
- USB port for communications with target hardware.
- Ethernet port for communications with target hardware (optional.

S32 Design Studio for S32 Platform 3.6.12 RFP Installation Guide specifies the installation prerequisites for Linux platforms.

2.4 Operational minimum configuration

Windows

- PC with 1.8 GHz Intel Pentium compatible processor.
- 4 GB of RAM.
- 25 GB of disk space.
- 20 GB of temporary storage (required only during the product installation).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- USB port for communications with target hardware.

Linux

- PC with 1.8 GHz Intel Pentium compatible processor.
- 4 GB of RAM.
- 20 GB of disk space.
- 16 GB of temporary storage (required only during the product installation).
- USB port for communications with target hardware.

2.5 Host operating system support

- Microsoft® Windows® 11 64-bit.

S32 Design Studio for S32 Platform 3.6.12 RFP supports all editions of the operating systems listed above and is limited only by the requirements of the Java Runtime Environment.

- Ubuntu LTS 24.04 64-bit

S32 Design Studio for S32 Platform 3.6.12 RFP supports all editions of the operating systems listed above and is limited only by the requirements of the Java runtime Environment.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

3 What's New

- Enabled **JTAG Daisy-Chain Debugging** support in the UI via a dedicated view, powered by the S32 Debugger backend running in the background. This enhancement enables seamless debugging across multiple connected targets, providing control directly within the UI, while simplifying issue diagnosis in complex chained-device setups.
- **S32DS Extensions and Updates** page detects when the update sites repository is not accessible due to network connectivity issues. In such cases, the UI now informs users about this, significantly reducing the time needed to investigate missing components.
- Expanded the availability of **NXP Application Code Hub** with more examples for the NXP FRDM development boards S32K3xx, S32M2xx to better support development, integration, and validation use cases.
- Bug fixes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

4 Known Issues

4.1 Known Issues

4.1.1 Breakpoints/Watchpoints Issues

Problem	Workaround
BBE32 DSP Add-On Package for S32R41 v1.2.0 cannot be updated/uninstalled: If BBE32 DSP Add-On Package for S32R41 v1.2.0 is installed in your IDE, it cannot be uninstalled or updated. If attempted to update/uninstall, this will break any installation for any package afterwards.	To fix this install a new S32DS IDE and install BBE32 DSP Add-On Package for S32R41 v1.2.1 or higher.
Install Remediation Page shows up when installing a package:	When installing a package from *S32DS Extensions and Updates*, you may see an *Install Remediation Page*, where you are asked to choose a solution to an installation problem. In most cases, choosing "*Update my installation to be compatible with the items being installed*" is the correct solution. If this is not giving you the right solution, try the other options, and inspect the solution details to make sure it is what you wanted.
Conditional watchpoint and breakpoints Conditional breakpoints and watchpoints, including those using ignore counts, may not work sometimes.	Avoid using conditions for breakpoints and watchpoints, instead check for condition in the code and set a normal breakpoint.
Watchpoints set on complex data types are not hit When debugging on a target connected with s32 debug probe, a debug session ignores a watchpoint set on a variable of a complex data type (such as a structure or other). a watchpoint set on an item of a basic data type inside a complex variable works correctly.	Avoid setting watchpoints at complex data types.
Duplicated error dialog When adding more watchpoints than supported by the device, the pop up box with the "not enough hardware resources for processing" error message is displayed twice.	Close the pop-up box, then close it again.
Step out work incorrect with dbg_derived_types source The ghs toolchain does not generate dwarf call frame information that is suitable for unwinding the call stack. For third-party debuggers incapable of unwinding the call stack through an alternate means (disassembling code), this may nullify the ability to walk the call stack or view variables saved in previous frames.	Set a breakpoint at function end and run to breakpoint. As a workaround s32ds removes the debug frames info to force the gdb client to recreate the frames. Sometimes this method may not work and lead to incorrect behavior for step out command.

4.1.2 VDK Issues

Problem	Workaround
Stepping over the try-catch block fails on a vdk When debugging a project on a vdk, an attempt to step over the try-catch block fails.	The issue is specific for projects compiled with NewLib. Recreate a project and select NewLib Nano as the library.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Problem	Workaround
Debugging on a VDK may fail Some errors may occur due to unexpected termination of a debug session or unsupported registers and watchpoint, and so on.	Run a debug session in Virtualizer Studio.

4.1.3 Linux Issues

Problem	Workaround
USB connection failure when debugging with S32 Debug Probe on Linux VM A debug session fails on a Linux virtual machine with s32 debug probe connected through usb. debugging with an S32 debug probe connected through Ethernet can be done successfully.	Connect the probe to USB. Set up the debug configuration to use the Ethernet connection rather than USB. Specify the virtual IP address of the probe in the connection settings of the debug configuration. To obtain the required value, run the following command: sudo ifconfig -a The displayed output includes a section for the virtual Ethernet interface created on Linux for communication with the probe over the physical USB link. The inet addr value in this section is a local IP address starting with "169.254". Take the HWaddr value in the section and convert the last two bytes into decimal notation. Append the resulting numbers to "169.254" to obtain the required virtual IP address.
Getting started page Should be displayed blank in Linux.	Install Webkit1 for GTK2 using the following command: sudo apt-get install libwebkitgtk-1.0-0
S32 Design Studio has multiuser install issue on Linux	To install the S32 Design Studio for multiple users, perform the following steps: • Under the user with sudo rights: – Install the S32 Design Studio to the /opt/NXP/S32DS.3.xfolder, – Change permission:sudo chmod -R 770/opt/NXP/S32DS.3.x – Run the S32 Design Studio from the /opt/NXP/S32DS.3.4 folder and close. – Change permission:sudo chmod 770 /dev/shm/cll_mutex sudo chmod -R 770 /opt/NXP/S32DS.3.x/eclipse/configuration – Add the second S32DS user without sudo rights to the first user group: sudo usermod -aG <sudo_user_name> <S32DS_user_name> – Switch the user to the <S32DS_user_name>, – Run the S32 Design Studio from the /opt/NXP/S32DS.3.xfolder.
Connecting via Telnet to the S32 Debug Probe configuration console fails When connecting multiple S32 Debug Probes simultaneously to a Linux host using USB, Telnet will only be able to connect to the first device.	There are two options to access the S32 Debug Probe configuration console:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Problem	Workaround
	• Connect using one or more serial port device files at: /dev/ ttyACM0(first probe), /dev/ ttyACM1(second probe), and so on, with a program such as screen. • Connect using Telnet by adding temporary IP route table entries to the Linux host for the S32 Debug Probes: – Identify the virtual IP address of the probes running the following command:sudo ifconfig -a The displayed output includes a section for the virtual Ethernet interface created on Linux for communication with the probe over the physical USB link. The inet addr value in this section is a local IP address starting with "169.254". Take the ether value in the section and convert the last two bytes into decimal notation. Append the resulting numbers to "169.254" to obtain the required virtual IP address. Example: ether value62:9F:B4:C6:D2:E3 translates to IP address 169.254.210.227 Identify the IP route matching each probe's local IP address using the IP route's src field. To list the existing IP routes, use the following command:ip route Example of S32 Debug probe route: 169.254.0.0/16 dev enx629fb4c6d2e3 proto kernel scope link src 169.254.186.146 metric 101 Add a new route by replacing the 169.254.0.0/16 subnet in the existing route with the calculated virtual IP address for each probe using the following command: sudo ip route add <new_ip_route> Example:Sudo ip route add 169.254.210.227 dev enx629fb4c6d2e3 proto kernel scope link src 169.254.186.146 metric 101 • Connect to S32 Debug probe using Telnet

4.1.4 Startup/Display Issues

Problem	Workaround
Quick Fix mechanism:	
In rare cases, an error may appear in the *Problems* view indicating that a required package is missing for the existing project, even though the package is already installed.	This error is harmless and can be safely deleted without affecting project functionality.
In rare cases, an error may appear in the *Problems* view indicating that a required package is missing for the existing project, and the available *Quick Fix* does not list any packages to install.	Manually install the missing package from *S32DS Extensions and Updates*. Once installed, the error will disappear.
In some cases, the New Project Wizard may incorrectly display the status of installed or missing toolchains/ compilers, or the status may not appear at all.	
The *Quick Fix* mechanism displayed in the *Problems* view for missing packages in the current project is not supported in all scenarios.	
Disassembly view content might be destroyed occasionally	Close the Disassembly view and open it again by selecting **Window** > **Show View** > **Disassembly** on the menu.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Problem	Workaround
Views are not updated instantly after executing a command in the GDB console	Use the step command to see refreshed views
The register variable is not shown in Memory View If the compiler locates the variable in a register the debugger cannot show it in the memory view. "view memory" selection for this variable generates the error message: "cannot view memory on and(counter)"	Information on which register is assigned for the variable can be found in: • the Disassembly view; • the error message in the Expression view. If "encounter" was added in the Expression view, then the error message contains the information like this: Failed to execute MI command: -data-evaluate-expression *(encounter) Error message from debugger back end: Address requested for identifier "counter" which is in register $r<reg_num>
Project with huge number of macros freezes S32DS IDE The huge count of macros is critical for indexing sources. Editing a file with many macros causes 100% cpu usage and freezes s32ds ide.	Manually tune the s32d.ini file by removing the following lines: -Xms256m -Xmx2048m
Design Studio becomes unresponsive On Ubuntu 24.04, Design Studio becomes unresponsive, when going to the Configuration tab, and then browsing between any other tab in the same window.	Filtering the content using the search bar of the Configuration tab restores smooth tab switching.

4.1.5 Installation/Uninstallation/Launch Issues

Problem	Workaround
S32 Debugger is missing after update to S32 Design Studio IDE 3.6.11 Starting with S32 Design Studio IDE version 3.6.11 the software components from the IDE platform were decoupled, and after updating to version 3.6.11 the platform will not contain the rest of the components.	When updating make sure to install all the needed components (e.g. S32 Debugger) in the installation page.
Creating project in the root of drive C:\ fails When creating a project in the root directory, the project wizard shows the "can't create directory: c:\project_name" error message. windows does not allow the users to create files in this location.	Create a subdirectory in the root directory (for example, `C:\Projects`) and specify this location in the project creation wizard.
Cannot install Eclipse p2 installation units on Ubuntu 20.04 distribution when "Always trust all authorities" or "Always trust all content" is selected For these cases, the "always trust everything confirmation" window becomes hidden and the user cannot access it. this is an Eclipse issue documented here: **https://github.com/eclipse-equinox/p2/issues/570**.	Use the "Select All" option in all windows instead of "Always trust all authorities" or "Always trust all content".
Launching S32Debug Group on Secure parts generate NPE error	Fill Password in S32Debugger Configuration.
Uninstallation of P&E drivers The p&e device drivers are not uninstalled with uninstallation of s32 design studio. After that, an attempt to uninstall them manually can cause errors	If you do not need the P&E drivers for other products and devices, uninstall them before the product uninstallation.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Problem	Workaround
S32 Debugger and/or S32 Trace no longer work after update fails due to "access denied" error	Restart S32 Design Studio and install the update again, as instructed in the dialog. Once the update completes successfully, functionality should be restored to normal.
When installing the platform package in s32ds extensions and updates after having used the s32 debugger or s32 trace, an error message may appear to state that access is denied to files in use. After this, the s32 debugger and s32 trace may no longer function	
Hyperlinks may not work	Set a different browser as the default one.
They may not work correctly when microsoft Edge (early versions) is configured as the default browser.	
Silent Mode installation:	
The uninstall mechanism is using the same logic as the installation, meaning if the installer was installed using silent mode, the uninstall will also be silent and it will not prompt the user for any confirmation.	
An error message is displayed when upgrading the platform from S32 Design Studio 3.6.10 to 3.6.11 from "S32DS Extensions and Updates".	Upgrade the platform directly from S32 Design Studio 3.6.11 installer.
The message is not harmful to the current session and user can continue the normal work after pressing OK button.	
MCP JSON RPC component is not automatically installed after upgrading from an older release.	Add the MCP JSON RPC update site and install the component manually.
When upgrading S32 Design Studio from an older release (e.g., S32DS 3.6.10 or earlier) to S32DS 3.6.11, the MCP JSON RPC component is not automatically installed as part of the upgrade process. MCP-related features will be unavailable until the component is manually installed.	Manually install the MCP JSON RPC component from the S32DS 3.6.11 update site: 1. Launch S32DS 3.6.11. 2. Go to **Help** → **Install New Software…**. 3. Click **Add…** and enter the S32DS 3.6.11 update site URL. 4. Select **MCP JSON RPC** from the list of available components. 5. Click **Next** and follow the installation wizard. 6. Restart S32DS when prompted.

4.1.6 Build Issues

Problem	Workaround
GHS project with/without SDK build failed after renaming of the project	For GHS projects with/without SDK, after renaming of the project activate clean and refresh procedures before building
Header files are copied to the project, and the rename process affects the paths. GHS build system believes that the files are not changed and does not update them during building - this is the GHS build system specific.	
GHS project build gives the "Unresolved inclusion" warning when including a standard C library	Not to break the C dialect specification, we recommend adding a path to include files manually: Open project properties, go to **C/C++ General** > **Preprocessor Include Paths, Macros, and so on,** > **CDT User Setting Entries** and add an **Include Directory** with the path to GHS folder, for example ${S32DS_GHS_PATH}\ansi, and the **Contains system headers** option selected.
GHS toolchain specific implementation does not contain a build-in spec detector.	

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Problem	Workaround
The headless build system gives the "Managed build system manifest file error"	To further avoid this error messages, install the GHS toolchain.
This error appears if the ghs toolchain is not installed. It is specific to the **Eclipse build system**, but it is not a real error and does not affect build success.	
Projects renaming out of the workspace is not supported	Use one of the following options: • Copy the project to the workspace (use file system options). • Import the project with copying files to the workspace. Change to another workspace with the project to rename (when using several workspaces)
S32DS cannot perform its custom rename procedure for a project outside the workspace used in the session. The default **Eclipse** renaming functionality may produce incorrect results, leading to issues with launch configurations and other related files.	
Setting PYTHONHOME and PYTHONPATH path is not supported	These two environment variables should not be set (user or system level) when working with S32 Design Studio.
The scenario in which a user sets the environment variables python home and python path is not supported, as it can lead to unexpected results	
Headless build stalls for several minutes after a successful build in CI environments	Add the -no-indexer flag to the headless build command to disable CDT indexing, for example: org.eclipse.cdt.managedbuilder.core.headlessbuild -no-indexer ...
When running headless S32DS commands in continuous integration builds, the Eclipse process (org.eclipse.cdt.managedbuilder.core.headlessbuild) stalls for several minutes after a successful build. Build artifacts (*.elf*, *.map*, *.bin*) are generated correctly, but the Eclipse process remains active before returning control to the makefile, which significantly increases the total build time. The delay is caused by CDT indexing triggered during execution of the CDT Managed Build headless application.	

4.1.7 Debugging Issues

Problem	Workaround
S32 Debugger program counter does not support HEX-value setting	When creating a **Debug Configuration**, at the **startup** tab use one of the following options: • Use a label or function name instead for the **Set program counter at** field in **runtime Options**, • Uncheck the **Set program counter at** and add the following commands to the **Run commands** field: set $pc=<HEX-value> -exec-until *<HEX-value>
Variables view and Expressions view do not display the values correctly	Use the **Details** number format in these views as it is the only format that will display the correct value in this context.
Debugging on S32 Debugger: variables view and expressions view may not display the value of some variables correctly if those variables have been used in macros.	
Debugging on S32 Debugger may fail	Close the **Variables** view or make it inactive for a critical part of code (until pointers are not initialized in code).
Variables view may pose a problem when entering some function with uninitialized pointers as it generates random access which may corrupt the debugger. sometimes the	

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Problem	Workaround
invalid access may be generated when debugging optimized code (ghs toolchain) where the pointer location shared with some variable.	
Only addresses added on the central core remain saved	Terminate the debug sessions on each core individually, don't terminate all.
When debugging, multicore launch and add some memory monitor addresses and terminate&relaunch the debug session, only addresses added on central core remain saved \	
When a debug session fails to terminate correctly, any subsequent debug session will fail to start and the **"Launching: Performing pre-launch check..."** task stalls forever in the *Progress* view. The **arm-none-eabi-gdb** process remains active and locks the project folder, causing S32 Design Studio to become unstable.	Manually terminate the stalled **arm-none-eabi-gdb** process using the system Task Manager (or the **kill** command on Linux/macOS), then launch a new debug session.

4.1.8 Importing Issues

Problem	Workaround
Import C/C++ Executable as S32DS project with Lauterbach	**To debug the imported ELF project,** manually fill in the required absolute paths for the following fields in the launch configuration: Configuration File, Initial Working Directory, and PRACTICE Script
when importing a C/c++ executable with lauterbach, the configuration file and practice script fields in the launch configuration are empty	
Import C/C++ Executable as S32DS project with Tasking	**To debug the imported ELF project,** manually fill in the required absolute path for the workspace field in the launch configuration
when importing a c/c++ executable with tasking, the workspace field in the launch configuration is empty	
Folders are not excluded if the SDK is migrated on a S32DS 3.4.1 imported project	Reload the SDK and exclude the old SDK folder from the build path after the migration to a newer version of the SDK. This would correctly exclude the backup folders.

4.1.9 DIAB Issues

Problem	Workaround
"Move to" line in DIAB might not work accordingly	Don't change PC in projects compiled with DIAB.
Debugging of Cortex A53 projects with Diab Compiler	Perform some stepping within the debug session.
Debug session stops with "no source available for "main() at". It is due to diab version 7.0.3 have some limitations on debugging. Debug info for c files isn't loaded in time. After stopping at the main function, several assembler commands must be executed before debug info will be loaded	

4.1.10 Trace Issues

Problem	Workaround
For M7 cores, during the trace collection some other packages might be generated outside the actual execution of the core.	

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

4.2 Problem Reporting Instructions

The S32 Design Studio for S32 Platform general issues is tracked through the S32DS Public NXP Community space: https://community.nxp.com/community/s32/s32ds.

For confidential cases and cases which cannot be publicly shared on the NXP Community follow the steps described here: https://community.nxp.com/t5/S32-Design-Studio/bd-p/s32ds.

For cases with PEmicro submit a support request: https://www.pemicro.com/support/.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

5 Acronyms

5.1 Acronyms

The following table shows the acronyms used in this document.

Table 1. Acronyms

Acronym	Description
ARM	Advanced RISC Machine
EWL	Embedded Warrior Library
GCC	GNU Compiler Collection
GDB	GNU Debugger
GB	Gigabyte
I/O	Input/output
IDE	Integrated Development Environment
MCU	Microcontroller Unit
RAM	Random Access Memory
RTD	Real-Time drivers
S32DS	S32 Design Studio for S32 Platform
SDK	Software Development Kit
VDK	Virtualizer Development Kit

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

6 Revision history

6.1 Revision History

Table 2. Revision history

Document ID	Release date	Description
RN00294 v.2.0	26 October 2026	Updates for S32 Design Studio 3.6.12 release
RN00294 v.1.3	29 September 2025	Updates for S32 Design Studio 3.6.4 release

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Tables

Tab. 1. Acronyms .....14 Tab. 2. Revision history 15

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Contents

1 Getting Started 1
1.1 Installation 2
2 Release Specifics 3
2.1 Release Content 3
2.2 Supported Features 3
2.3 Recommended configuration 3
2.4 Operational minimum configuration 3
2.5 Host operating system support 4
3 What's New 5
4 Known Issues 6
4.1 Known Issues 6
4.1.1 Breakpoints/Watchpoints Issues 6
4.1.2 VDK Issues 6
4.1.3 Linux Issues 7
4.1.4 Startup/Display Issues 8
4.1.5 Installation/Uninstallation/Launch Issues 9
4.1.6 Build Issues 10
4.1.7 Debugging Issues 11
4.1.8 Importing Issues 12
4.1.9 DIAB Issues 12
4.1.10 Trace Issues 12
4.2 Problem Reporting Instructions 13
5 Acronyms 14
5.1 Acronyms 14
6 Revision history 15
6.1 Revision History 15
Legal information 16

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2026 NXP B.V. **All rights reserved.**