



UnderstandTM
by SciTools

Safety Manual

Build 7.2.1248

Prepared by Kylor Kersavage

For safety related questions please contact support@scitools.com

Revision History

Author	Approved By	Date	Reason For Changes	Version
Kylor Kersavage	Heidi Esplin	10/18/22	First Draft	1.0
Kylor Kersavage	Heidi Esplin	03/15/2023	Updated tool version Changed ISO spec in sec. 2 to :2018 sec. 4 rewritten, added approved languages and OSs. sec. 9 added crash handling and user popups	1.1
Kylor Kersavage	Heidi Esplin	12/18/2025	Updated tool version to 1239, updated operating constraints.	1.2
Kylor Kersavage	Kevin Groke	04/07/2026	Added additional use cases, added TCL determination section, added ASIL applicability, added qualification work products, added safety related requirements, section 4 updated, added verification and traceability to section 8. Updated build number to 1248.	2.0

1 **TABLE OF CONTENTS**

2	Tool Version	4
3	Tool Description	4
4	Certified Pre-Qualification	4
5	Other applicable User Documentation	4
6	Tool usage and flow in typical design environments	4
7	Use Cases	5
8	Tool Classification (summary)	7
9	Assumptions of Use for ISO 26262 compliance	8

2 TOOL VERSION

Tool Name: Understand

Tool Version and ID: Version 7.2.1248

3 TOOL DESCRIPTION

Understand is a static code analysis tool that assists with editing, refactoring, and analyzing code. This tool can be used to assist with the development of safety critical automotive systems as defined in ISO 26262:2018.

4 TOOL CERTIFICATION

This tool has been certified according to ISO 26262 and IEC 61508 for the version identified in this Safety Manual. A list of certified use cases can be found in section 7. Understand is certified for safe use with the following languages:

C
C++

Understand is certified for safe use with the following Operating Systems:

Windows x64 (Windows 10.1809 and later),
Linux 64bit (CentOS and RHEL 8.6 (4.18.0-372.26.1 or newer), Ubuntu 22.04, and later)
Mac OS (Monterey 12 and later) Intel and Apple Silicon are both supported

NOTE: Update your OS to the latest version for the safest experience

Standards Applicability:

The tool qualification activities documented in this Safety Manual are performed in accordance with ISO 26262-8. These activities and the associated qualification evidence are also considered sufficient to satisfy the applicable software tool qualification requirements of IEC 61508 for the intended use. No additional IEC 61508-specific tool qualification activities are required.

5 OTHER APPLICABLE USER DOCUMENTATION

The Understand User Guide can be found within the Understand Software by selecting Help > Help Content from the main Toolbar.

Additional user resources including videos, build notes, known issues, and the installation guide can be found at support.scitools.com.

For use in a safety context make sure to follow the section of the Help Content titled “Configure your Project”. Proper project configuration is essential for accurate analysis.

6 TOOL USAGE AND FLOW IN TYPICAL DESIGN ENVIRONMENTS

6.1 OPERATING CONFIGURATION

A valid license must first be obtained before importing source code into Understand. Please contact sales@scitools.com for more information.

Refer to the “Configuring Your Project” section of our help guide to see detailed setup instructions for the programming language you plan on analyzing. The help guide can be found within Understand or accessed on our support site [here](#).

Your local development environment needs to be configured appropriately before running Understand. An improper setup will result in inaccurate information. The Analysis Log will display errors or warnings when importing your code base in the case of misconfiguration. Resolve all errors or warnings to receive the most accurate information about your code.

6.1.1 Optional Understands Settings

Tools > Options > Analyze > Automatically analyze changed files on save

By default Understand will only analyze a file when the command is given. Turn this option on to have an analysis run automatically every time a file is changed and saved.

Analysis, metrics, and graphs will not reflect user updates until a new analysis has been performed. An analysis can be run manually at any time by going to Project > Analyze Changed Files.

CodeCheck will ask the user to perform an analysis when changes are detected.

6.2 OPERATING CONSTRAINTS

The machine that is running the Understand application should have 1Gb RAM for every 1 million lines of source code that is being statically analyzed. The source code can be on a network, however the database that the Understand application generates, should be on the local machine for optimal performance. This database is generated when a project is created in Understand and an analysis has been run.

The general licensing of Understand does require an internet connection. The Understand app communicates with our license server (licensing.scitools.com) on a regular basis. This information is secured using SSL and this information is not shared and is only used for licensing purposes. In some cases, networks/firewalls may need to be configured to allow access on TCP port 443 using secure HTTPS. Firewalls with application level security may need to enable HTTPS permission for the Understand, und, and uPython executables installed in SciTools/bin/directory.

For licensing that requires zero internet connection, there is a closed lab proprietary license server that can be used, Icarus, which requires setup by a user with admin privileges and defaults to ports 80 and 443. Icarus only supports 64 bit operating systems. A guide for setting up an offline server license can be found [here](#).

6.3 Operating Environment

Disk Space: At least 2GB free

CPU: Any modern CPU

RAM: 1GB of Ram Per Million lines of source code

Operating System: Understand will run on:

- Windows x64 (Windows 10.1809 and later),
- Linux 64bit (CentOS and RHEL 8.6 (4.18.0-372.26.1 or newer), Ubuntu 22.04, and later)
- Mac OS (Monterey 12 and later) Intel and Apple Silicon are both supported

7 USE CASES

Tool	Use Case	Specification	References to user documentation
New Project Wizard	Manual Code Import	Create a new Understand project by selecting the top-level directory where your source code resides.	Help Guide > Configuring Your Project > Creating a New Project
New Project Wizard	Cmake code import	Create a new Understand Project by allowing Understand to access your compile commands document generated by Cmake.	Help Guide > Configuring Your Project > Creating a New Project > Creating Projects from Cmake Projects
New Project Wizard	Buildspy code import	Create a new Understand Project by selecting the buildspy option, selecting your local source code directory, and rebuilding your source code while Buildspy watches.	Help Guide > Configuring Your Project > Creating a New Project > Using Buildspy to Create Projects
Architecture Designer	Create custom architecture	Design your own tree graph showing the interactions of your code. This is done by creating nodes and sub nodes that relate to each other.	Help Guide > Architecting Your Codebase > Creating and Editing Custom Architectures
CodeCheck	Analyze Code	Select a variety of checks to run against your source code. These checks can look for a combination of formatting, syntax, and best practice errors. A list of the errors is displayed to the user and can be exported and shared.	Help Guide > Using CodeCheck for Standards Verification > Running a Code Check
Project Analysis	Error Reporting	Analyze the user's source code looking for syntax errors.	Help Guide > Configuring Your Project > Analyzing the Code
Architecture Browser	Import Architecture	Import an architecture saved to the user's computer and display it for the user.	Help Guide > Architecting Your Codebase > Sharing Architectures

Architecture Browser	Export Architecture	Export an architecture by saving a file to the user's computer.	Help Guide > Architecting Your Codebase > Sharing Architectures
Architecture Browser	Navigate Architectures	Navigate a custom architecture by clicking through the nodes and sub nodes in a text-based tree hierarchy.	Help Guide > Architecting Your Codebase > Browsing Architectures
Graphing Tool	Architecture Dependency Graph	View the structure of an architecture in a tree graph format.	Help Guide > Architecting Your Codebase > Viewing Architecture Graphs > Dependency Graphs
Graphing Tool	Directory Dependency Graph	View relationships between directories of code in a tree graph structure.	Help Guide > Using Graphical Views > Opening Graphs
Graphing Tool	File Dependency Graph	View relationships between files in a tree graph structure.	Help Guide > Using Graphical Views > Opening Graphs
Graphing Tool	Metrics Graph	Displays files as blocks sized by line count and changes shades of blue based on selected metric.	Help Guide > Using Metrics > Metrics Tree map
Graphing Tool	Calls/Called By Tree Graph	Creates a tree graph showing the entities that are called by the selected item and the entities that call the selected item. This graph can be used for functions, files, or directories.	Help Guide > Using Graphical Views > Opening Graphs
Metrics Browser	View Metrics	Show relevant metrics for a selected entity. Metrics can be shown for functions, files, and directories and are displayed in a partitioned window call the Metrics Browser.	Help Guide > Using Metrics > Metrics Browser
Metrics Browser	Export Metrics	Export the currently shown metrics via a button in the Metrics Browser.	Help Guide > Using Metrics > Export Metrics

8 TOOL CLASSIFICATION (SUMMARY)

The following use cases are classified as TI2, meaning they could potentially impact executable code. Refer to the table below for possible malfunctions and mitigations.

Tool	Use Case	Malfunction	Cause (opt)	Effect on item or element	TI	If TI2: Existing Protective / Detection measure	Resulting TCL
New Project Wizard	Manual Code Import	Empty Project is generated	Unsupported code files or file extensions	Project Analysis will be unable to provide meaningful output	TI2	Project Analysis Reports Errors/Warnings	TCL1
New Project Wizard	Manual Code Import	Project definition is not complete	Missing header files	Project Analysis will produce errors and warnings	TI2	Project Analysis Reports Errors/Warnings	TCL1
New Project Wizard	Cmake code import	Project definition is not complete	Build fails to complete	Analysis data will be missing for unanalyzed code	TI2	Project Analysis Reports Errors/Warnings	TCL1
New Project Wizard	Buildspy code import	Project definition is not complete	Build fails to complete	Analysis data will be missing for unanalyzed code	TI2	Project Analysis Reports Errors/Warnings	TCL1
Architecture Designer	Create Custom Architecture	Architecture Designer doesn't save changes to architecture	Understand couldn't update Architecture file	None	TI2	Failure to write architecture is reported to user via popup notification	TCL2
CodeCheck	Analyze Code	Script Error	Perl/Python script crashed during execution	Indicated script failed to check indicated file	TI2	Scripting error reported to CodeCheck log	TCL1
Project Analysis	Error Reporting	Analysis errors while analyzing the project	Missing header files	Project database will have inaccurate/incomplete data	TI2	Find Missing Includes wizard	TCL1
Project Analysis	Error Reporting	Analysis errors while analyzing the project	Missing macro definitions	Project database will have inaccurate/incomplete data	TI2	Analysis errors reported to users	TCL1

Project Analysis	Error Reporting	Analysis errors while analyzing the project	Unfamiliar syntax	Project database will have inaccurate/incomplete data	TI2	Analysis errors reported to users	TCL1
Project Analysis	Error Reporting	Analysis errors while analyzing the project	Source files corrupted or unreadable	Project database will have inaccurate/incomplete data	TI2	Compilation units with corrupted or unreadable source files will have incomplete analysis data	TCL1

TOOL CONFIDENCE LEVEL (TCL) DETERMINATION:

The failure modes listed above represent the qualification-relevant use cases. The Architecture Designer failure mode (“Create Custom Architecture – failure to save changes”) results in TI2 with TD2, and therefore determines the maximum applicable Tool Confidence Level (TCL 2) for the intended use. The complete failure mode analysis and justification of TI/TD/TCL assignments is documented in the tool qualification FMEA.

VERIFICATION AND TRACEABILITY

The mitigations described in this section are verified through documented regression and verification activities for the certified tool version. Each mitigation is traceable to specific test cases and executed test results as documented in the tool qualification FMEA, which links the defined detection measures to the corresponding verification activities and evidence for the certified build.

ASIL APPLICABILITY:

Under the assumptions and constraints described in this Safety Manual, the qualified use of Understand is applicable for safety-related development activities up to ASIL D in accordance with ISO 26262-8.

QUALIFICATION WORK PRODUCTS

The tool qualification is supported by a defined set of qualification work products, including documented failure mode analysis, change impact assessment, and verification evidence for the certified build. These work products together provide the complete justification of the tool qualification in accordance with ISO 26262-8 and are maintained as part of the certification package.

9 CONDITIONS AND ASSUMPTIONS OF USE FOR ISO 26262 COMPLIANCE

9.1 ASSUMPTIONS AND LIMITATIONS OF USE

Error Detection: Most Issues will be reported to the user via the “Analysis Log”. An analysis will be run automatically after importing code. You can also run an analysis manually with “Ctrl + Alt + R”.

Crash Detection and Handling: When an exception is encountered that cannot be resolved Understand will close the application and present the user with a crash notification. All currently running operations are canceled and the data is reverted to the last successful save. Make sure to check your data in the case it has been reverted to a previous version.

CodeCheck: Issues with the CodeCheck Scripts are reported within the CodeCheck window under “Script Errors”. If any script errors are detected the CodeCheck results may be invalid.

Sudden Interruptions: Anything that disrupts an operation in Understand will cause that operation to be aborted. You will need to restart the task. This includes power failures, operating system failures, Understand critical failure “crash”, etc.

User Suggestions (pop-ups): Understand will detect when a possibility of incorrect data arises. For example, if an analysis is requested, Understand will detect unsaved changes and suggest the user saves before running the analysis. It is recommended that all popups and notifications are read before proceeding with an action.

9.2 SAFETY RELATED REQUIREMENTS FOR USE

To ensure the safe use of the tool within a safety-critical development process, the following requirements shall be observed by the user:

- The tool shall be used only for the intended use described in this Safety Manual.
- Tool-reported errors, warnings, or abnormal terminations shall be treated as failures and investigated before relying on the results.
- Generated outputs (e.g., architecture artifacts, analysis results, reports) shall be reviewed for plausibility and completeness before being used as safety-relevant inputs.
- Verification or regression activities defined by the user’s development process shall be successfully completed for the tool version in use.
- The user shall ensure that any tool configuration or automation scripts used are appropriate for the intended use and environment.