Library Compiler™
Functional Safety Manual

June 2018, Revision 1.5

SYNOPSYS®

Copyright and Proprietary Information Notice

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA, 94043
www.synopsys.com

Document Control

Revision history

Version	Description	Date
1.0	First release of the document submitted for initial review.	28-Jan-2018
1.1	Added revision history, fixed template issues.	01-Feb-2018
1.2	Fixed boilerplate changes from general feedback.	01-Mar-2018
1.3	Updated AoUs	18-May-2018
1.4	Updated AoUs	30-May-2018
1.5	Added AoU-LC-007	4-Jun-2018

Contents

1
Customer Support

This section describes the customer support that is available through the Synopsys SolvNet® customer support website or by contacting the Synopsys support center.

Accessing SolvNet

The SolvNet support site includes an electronic knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. The site also gives you access to a wide range of Synopsys online services, which include downloading software, viewing documentation, and entering a call to the Support Center.

To access the SolvNet site:

1. Go to the web page at https://solvnet.synopsys.com.
2. If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register.)

If you need help using the site, click **Help** on the menu bar.

Contacting Synopsys Support

If you have problems, questions, or suggestions, you can contact the Synopsys support center in the following ways:

- Go to the Synopsys Global Support Centers site on synopsys.com. There you can find e-mail addresses and telephone numbers for Synopsys support centers throughout the world.
- Go to either the Synopsys SolvNet site or the Synopsys Global Support Centers site and open a case online (Synopsys user name and password required).

2
Scope of This Document Scope of This Document

This section describes the scope of this document and defines terms used in this document.

Using This Document

The *Library Compiler Functional Safety Manual* describes the proper use of the Library Compiler tool in safety-related applications according to the ISO 26262 standard, and is intended to confirm the compliance of the Library Compiler tool to the standard when used in the context of a tool chain.

The Library Compiler tool reads the description of a logic library from a text file (.lib file format) and compiles the description into an internal database (.db file format). The compiled database supports synthesis, place and route, static timing analysis, and verification tools.

Section 3 describes an overview of the ISO 26262-8, clause 11 and the approach adopted by Synopsys to comply with the requirements of the standard. Section 4 defines the general information such as where to find the latest documentation and installation requirements regarding the use of the Library Compiler tool as a software tool in the development of safety-related applications. Section 5 shows the high-level overview of the tool chain that this product belongs to. Section 6 details the safety-related requirements for safety-qualified use cases of the Library Compiler tool. Section 7 lists the known limitations of the use cases.

Specific documentation for performing design and analysis as part of an ISO 26262 compliant flow is provided in Section 3, Section 5, Section 6 and Appendix A of this document, the *Library Compiler Functional Safety Manual*.

Terms and Definitions

Term	Definition
.clib	A file format to store cell library database that may also include the physical views Combines the logical and physical information for a cell; also known as “reference library”
.db	A binary file format to store library data.
.lib	Synopsys text file format to store library data.

Term	**Definition**
AoU	Assumption of Use. An action that is assumed and required to be taken by the user of a software tool.
ASIL	Automotive Safety Integrity Level. This is a risk classification scheme defined by the standard ISO 26262. The standard identifies four levels: ASIL A, ASIL B, ASIL C, and ASIL D. ASIL D dictates the highest integrity requirements on a product and ASIL A dictates the lowest.
AWP	Advanced waveform propagation A Synopsys technology for use in advanced process nodes to optimize and analyze waveform distortions, such as in static timing analysis. This feature requires both the CCS timing and the CCS noise library models.
CCS	Composite current source Liberty advanced library models for timing, noise, and power
CEL	The name of the view in the Milkyway interface that contains routing information such as layers, pins, and connectivity and that represents the routed data.
Component	A part of an electronic system that implements a function in a vehicle. See also Part 1 of the standard ISO 26262 for the definition. The standard also refers to elements and items, but for the *Library Compiler Functional Safety Manual*, there is no difference.
CoU	Condition of Use. A condition of the design, software tool, design environment, or situation that is assumed and required to be fulfilled by the user.
CRM	Customer Relationship Management. Internal Synopsys database that manages customer STARs.
Defect	Product nonconformance.

Term	**Definition**
Error	An error is a discrepancy between the actual and the specified or theoretically correct operation of an element. The root causes of an error can be manifold. In this document, the focus is on errors that are introduced or left undetected in a design, due to the malfunction in a software tool (e.g. generation of bad logic by a logic synthesis tool, failure of a static timing analysis tool to detect a timing violation).
Fault	An abnormal condition that can cause an element or item to fail.
Fault analysis	An analysis that determines the behavior of a system when a fault is introduced.
FMEA	Failure Mode and Effects Analysis. An analysis that looks at different parts of a system, identifies ways the parts could fail, and determines the causes and effects of these potential failures.
FRAM	The name of the frame view in the Milkyway interface that is an abstraction of the CEL view used by the router and that contains pins and obstruction information.
GDS	Graphic Database System A database file format that is a standard for open artwork interchange.
LC	Abbreviation for Library Compiler
LEF	Library Exchange Format Standard for representing physical layout of an integrated circuit.
Liberty	A gate-level modeling standard for logic libraries including timing, noise, power and test behavior consumed by design implementation, sign-off, and verification tools.
Library Compiler	The tool name
Logic library	A library that includes timing, functionality, power, and noise information for the available components. The technology of each library is specific to a particular ASIC vendor. The technology of a given library is determined by the cell function and attributes contained in the library.
MCMM	Multicorner-multimode

Term	Definition
Milkyway™	The Milkyway interface is a Synopsys tool for preparing physical libraries from the layout data.
OASIS	Open Artwork System Interchange Standard
Physical library	A library that includes physical layout information for library cells mainly used by physical implementation tools for placement and routing.
PVT	Process, voltage, temperature for representing the operating conditions of an integrated circuit
Software / software tool	The Library Compiler tool.
Software tool criteria evaluation	Analysis according to ISO 26262 to determine the required TCL of a software tool.
Software tool qualification	Means to create evidence, that a software tool with low or medium TCL is suitable to be used in the development of safety related products according to ISO 26262.
SolvNet	Synopsys customer support site.
Standard	In this document, refers to *ISO 26262 Road Vehicles – Functional Safety*, 2011 and 2018 versions.
STAR	Synopsys Technical Action Request. A STAR documents and tracks a product Bug or Enhancement request (called a B-STAR or an E-STAR, respectively). It is stored in the Synopsys CRM database. Only Synopsys employees can access the CRM database. However, limited STAR information is available from SolvNet for customers who are associated with the user site of a STAR. Customer contacts are notified automatically when a STAR is filed or when its status changes.
TCL	Tool confidence level, as defined by ISO 26262-8, clause 11. Note: The TCL of a software tool does not necessarily indicate whether the tool may malfunction or not. The TCL defines the confidence level that an error in the safety-related design, which is introduced or left undetected by the software tool, can be prevented or detected in subsequent steps of the development flow, before the erroneous safety-related design is released.
TD	Tool error detection, as defined in ISO 26262-8, clause 11.
TI	Tool impact, as defined in ISO 26262-8, clause 11.

Term	**Definition**
UPF	IEEE 1801, also known as the Unified Power Format
Use case	A use case is a specific way of using a software tool, that can be characterized by: - a limited set of tool functions and features that are used; - a set of restrictions and constraints that are regarded while using the tool; and - a specific goal to be achieved or output to be generated by using the software tool Use cases may be associated with different steps or phases in the design process, or they may describe alternative ways of using the tool for a specific design step.

3 Confidence in the Use of Software Tools According to ISO 26262-8, Clause 11

This section provides an overview of the ISO 26262-8, clause 11. It then describes the approach adopted by Synopsys to comply with the requirements of the standard, and how this is mapped to activities performed by Synopsys and the end user of the Synopsys tools.

Overview of ISO 26262-8, Clause 11

Synopsys EDA software tools contribute significantly to the design specification, implementation, integration, verification and validation of electrical and electronic (E/E) systems and components. If these E/E systems and components are used as part of a safety-related automotive product, an error in these systems or components could have severe consequences on functional safety. Such an error may arise as a result of unforeseen operating conditions or due to a fault introduced during product development, which in turn may be caused by a software tool malfunction. ISO 26262-8, clause 11 (Confidence in the Use of Software Tools) addresses this issue and specifies requirements and methods which aim to minimize the risk of faults in the developed product due to malfunctions of a software tool affecting the product’s functional safety.

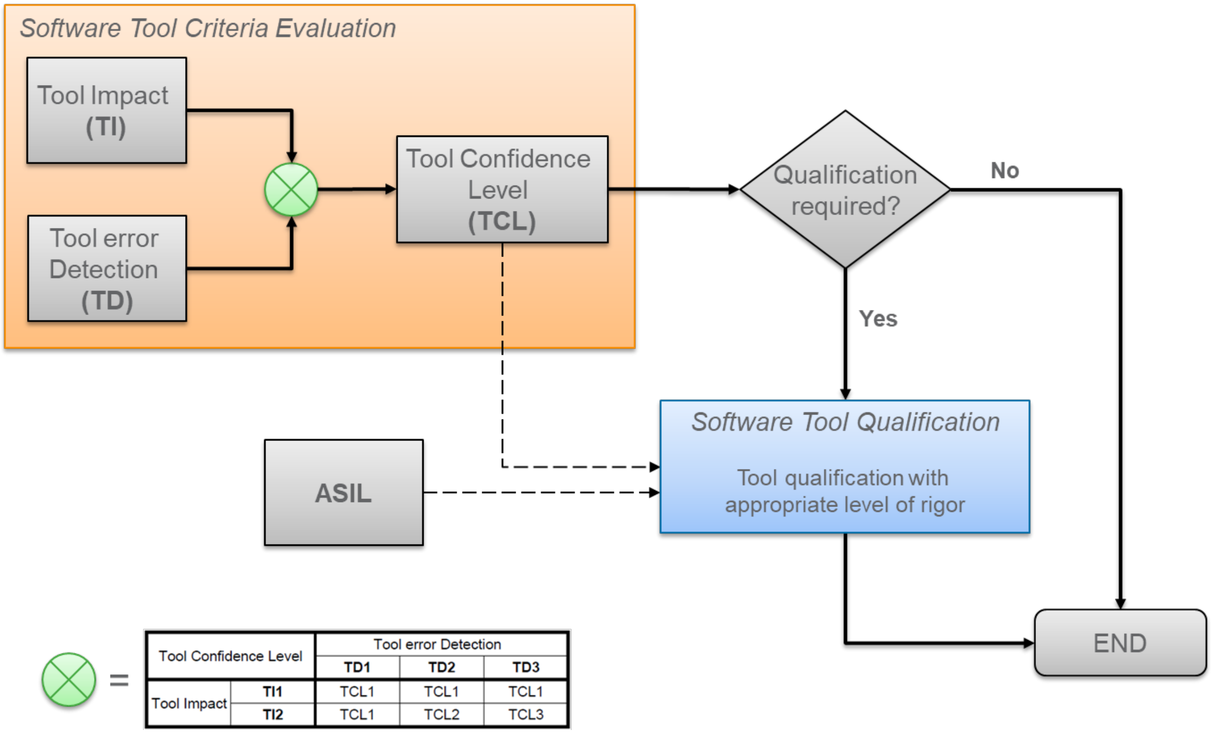
According to ISO 26262, to determine the required level of confidence in a software tool that is used in the development of a safety-related automotive product, the following criteria are evaluated:

- The possibility that the malfunctioning software tool and its corresponding erroneous output can introduce or fail to detect errors in a safety-related element being developed.
- The confidence in preventing or detecting such errors in its corresponding output.

This procedure is called Software Tool Criteria Evaluation, and it must be performed for all software tools that are involved in the development of a safety-related element, resulting in a required Tool Confidence Level (TCL) for each software tool.

If the software tool criteria evaluation determines that a medium or high TCL is required, then appropriate Software Qualification methods must be applied, effectively reducing the risk of a critical software tool error. The choice of software qualification methods depends on the required TCL and the maximum ASIL of all the safety requirements allocated to the element developed using the software tool. However, if the software tool criteria evaluation determines that only a low TCL is required, then there is no need to apply such software qualification methods.

The software tool criteria evaluation and software tool qualification flow is summarized in Figure 1.


Software Tool Criteria Evaluation
Tool Impact (TI)
Tool error Detection (TD)
Tool Confidence Level (TCL)
Qualification required?
No
Yes
Software Tool Qualification
Tool qualification with appropriate level of rigor
ASIL
END
=

Tool Confidence Level		Tool error Detection		
		TD1	TD2	TD3
Tool Impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Figure 1: Software tool criteria evaluation and software tool qualification flow

Work Split between Synopsys and Tool Users

A software tool criteria evaluation must always be performed in the development environment of the final tool user, and in the context of the actual product development. It is in this context, where potential tool malfunctions, their effect on the safety-related product, and the effectiveness of prevention and detection measures must be analyzed.

However, the tool vendor can support the tool user by performing a software tool criteria evaluation (and, if required, a software tool qualification) on their own, based on assumed tool use cases and an assumed development environment. If the assumptions made by the tool vendor match the actual situation at the tool user, then the user can take over the evaluation (and qualification) results from the tool vendor. Besides significantly reducing the effort for the tool user, this approach can also result in a better quality for the software tool criteria evaluation and qualification, since the tool vendor typically has a more detailed understanding of the inner working and possible malfunctions of the software tool.

Synopsys has adopted exactly this approach, which is summarized in Figure 2.

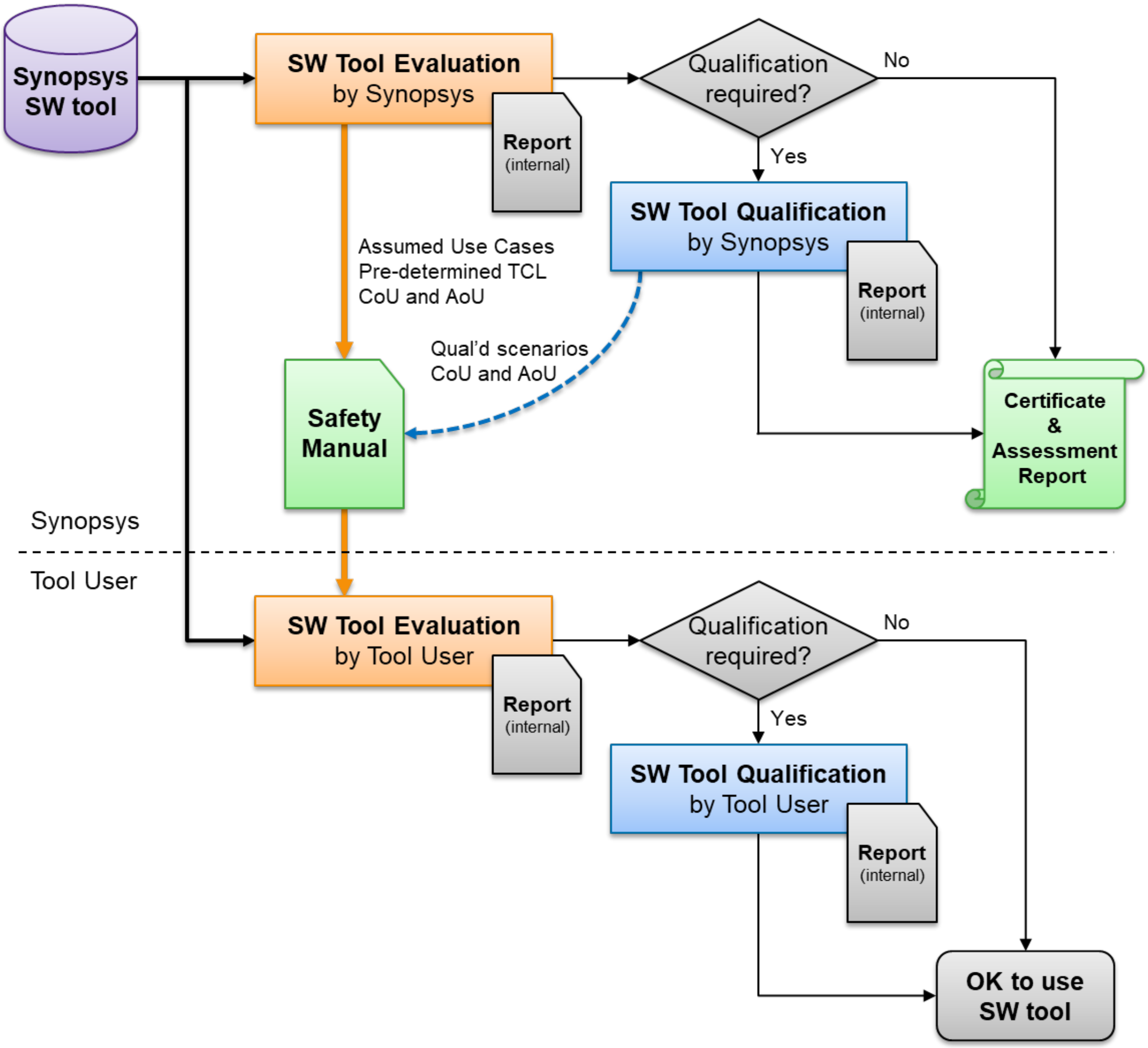


Figure 2: Work Split between Synopsys and Tool Users

Synopsys performs the following activities:

1. Software tool criteria evaluation
   - Identification of possible **use cases** for the software tool, together with required **inputs** and expected **outputs**
   - Specification of **conditions of use (CoU)** for each use case, related to the development environment in which the tool is assumed to be deployed, including tool usage procedures and constraints
   - Analysis of potential software tool **malfunctions**, and their effect on a safety-related product that is developed with this tool
   - Analysis of **prevention** and **detection measures** internal to the software tool, to avoid tool malfunctions, or to control and mitigate their effects
   - Specification of **assumptions of use (AoU)**, which are additional prevention and detection measures assumed to be performed by the end user of the tool

- Estimation of the **Tool Impact (TI)** for each malfunction, and the probability of **Tool error Detection (TD)** by the prevention and detection mechanisms (including assumptions of use)
- Determination of the required **Tool Confidence Level (TCL)** for each software tool malfunction, based on TI and TD
- Determination of the maximum TCL from all software tool malfunctions related to a use case. This is called the **pre-determined TCL** for the software tool use case
- Summary of the results in a software tool criteria evaluation report

2. Software tool qualification
   - If the pre-determined TCL indicates, that a medium (TCL2) or high (TCL3) tool confidence level is required for the software tool, then Synopsys may decide to perform a software tool qualification
   - The specific methods applied for tool qualification can vary for different tools and use cases, and they may include an evaluation of the software tool development process, the validation of the complete software tool, the validation of critical tool malfunctions with insufficient prevention and detection measures, or other methods
   - Summary of the qualification methods, procedures and results in a software tool qualification report

3. Safety manual for the software tool
   - The *Library Compiler Functional Safety Manual* (this document) is an important deliverable to the tool users, as it includes all end user-relevant information from the Synopsys software tool criteria evaluation and qualification
   - Software tool criteria evaluation related information, documented in Section 6, includes:
     - Description of software tool use cases
     - Description of the required inputs and expected outputs for each use case
     - Specification of conditions of use (CoU – conditions of the design, software tool, design environment, or situation that are assumed and required to be fulfilled by the user) for each use case
     - Specification of assumptions of use (AoU – actions that are assumed and required to be taken by the user of a software tool) for each use case
     - Pre-determined TCL for each use case
   - Software tool qualification related information (not required for this Library Compiler and therefore not included in this safety manual)
     - Description of the scope of the software tool qualification, including malfunctions and scenarios covered by the qualification
     - Specification of additional conditions of use (CoU) derived from the software tool qualification
     - Specification of additional assumptions of use (AoU) derived from the software tool qualification
   - Other information included in this safety manual
     - General information about the software tool needed by the tool user (see Appendix A)
     - Known limitations of the software tool, related to the described use cases as documented in Section 7

4. Certification and assessment report
    - Synopsys may decide to perform a functional safety assessment, to confirm the correctness, completeness and ISO 26262 conformance of the performed software tool criteria evaluation and qualification
    - Synopsys may also decide to achieve certification from an accredited third-party certification body, in addition to the functional safety assessment
    - The results of these activities are summarized in a functional safety assessment report and a certificate which can be viewed at exida Certificate for ISO 26262 Compliance

If the tool user wants to benefit from the work done by Synopsys, then according to the Figure 2 above, the user shall perform the following activities for each software tool:

1. Software tool criteria evaluation
    - Review and verify that the software tool criteria evaluation (and qualification) performed by Synopsys, as documented in the tool's Functional Safety Manual, matches the actual situation of the user's product development process
        - Verify whether the actual use case(s) of the software tool match those evaluated by Synopsys
        - Verify whether the actual inputs and outputs are identical to or a sub-set of those as evaluated by Synopsys
        - Verify that all conditions of use (CoU) specified by Synopsys are met, or whether the development process can be adjusted to meet these CoU(s)
        - Verify that all assumptions of use (AoU) specified by Synopsys are met, or whether the development process can be adjusted to meet these AoU(s)
        - Verify that the pre-determined Tool Confidence Level (TCL) for the relevant use case(s) are TCL1, *or*
        - Verify that Synopsys has successfully performed an additional software tool qualification for all TCL2 and TCL3 scenarios to conclude that the tool is suitable to be used for the development of a safety-related element of the same or higher ASIL than required by the user
    - If all the verification steps described above are successful, then the results of the Synopsys software tool criterial evaluation (and qualification) are applicable to the tool user, which means:
        - The required TCL pre-determined by Synopsys can be taken over by the tool user for actual product development
        - If the pre-determined TCL is TCL1, then the tool can be used without the need to perform any additional software tool qualification
        - If the pre-determined TCL is TCL2 or TCL3, then the software tool qualification performed by Synopsys is sufficient, and the tool can be used without the need for further software tool qualification by the end user
    - All of the steps above must be documented in a software tool criteria evaluation report, including evidence for the successful conclusion of all verification steps, which may include reference to the Synopsys Functional Safety Manual, and optionally, to the Synopsys certification and assessment report

2. Software tool qualification
    - If any of the verification steps described above as part of the tool user’s software tool criteria evaluation fails (e.g. different use case, CoU or AoU cannot be met, pre-determined TCL is not TCL1 and Synopsys has not performed a software tool qualification), then the user must perform his/her own software tool qualification
    - The specific methods applied for tool qualification are decided and planned by the tool user -- Synopsys does not recommend any specific methods or procedures
    - The summary of the qualification methods, procedures and results shall be documented in a software tool qualification report

4 Library Compiler Description

This section provides a general description regarding the use of the Library Compiler tool as a software tool in the development of safety-related applications and describes where to get the latest product documentation and the runtime environment required to use the Library Compiler tool.

Coverage

The *Library Compiler Functional Safety Manual* is intended to be used starting with the version 2017.12 and later versions of the Library Compiler tool per the use cases presented in this document. In general, unless otherwise noted, the failure modes and detection mechanisms noted in the use cases presented in Section 6 are tool version independent.

Compliance with ISO 26262

The Library Compiler tool can be used in the development of safety-related elements according to ISO 26262, with allocated safety requirements up to a maximum Automotive Safety Integrity Level D (ASIL D), if the tool is used in the context of a tool chain and in compliance with this document, the *Library Compiler Functional Safety Manual*.

See the exida Certificate for ISO 26262 Compliance of Synopsys Library Compiler when used in a tool chain flow.

Product Documentation and Support

Comprehensive documentation for using the Library Compiler tool is provided on SolvNet. The latest documentation for the Library Compiler tool can be accessed at Library Compiler Online Help on SolvNet.

.

Specific documentation for performing design and analysis as part of an ISO 26262 compliant flow is provided in Section 3, Section 5, Section 6 and Appendix A of this document, the *Library Compiler Functional Safety Manual*.

Synopsys provides online customer support for the Library Compiler tool. See Section 1 for more information.

Installation and Supported Platforms

The installation of the Library Compiler tool must follow the guidelines in the *Synopsys® Installation Guide* as well as the specific *Library Compiler Installation Notes* document.

Users are required to download the tool executable and INSTALL_README from the SolvNet site at https://solvnet.synopsys.com/DownloadCenter/dc/product.jsp.

Supported platforms and operating systems requirements:

- For installation instructions, see the *Synopsys® Installation Guide* at https://www.synopsys.com/install.
- For the latest supported binary-compatible hardware platform or operating system, including required operating system patches, see https://www.synopsys.com/qsc.
- If updates (including security patches) to computing environments (including operating systems) are backward compatible with previous versions of the computing environment used to test the Library Compiler tool, the results of the testing performed by Synopsys using such previous versions are applicable.

Additional information:

- For information about the compute platforms roadmap, go to https://www.synopsys.com/support/licensing-installation-computeplatforms/compute-platforms/compute-platforms-roadmap.html.
- For platform notices, go to https://www.synopsys.com/support/licensing-installation-computeplatforms/compute-platforms/platform-notice.html.
- For information regarding the license key retrieval process, go to https://solvnet.synopsys.com/smartkeys/smartkeys.cgi.

User Competence

To properly use the Library Compiler tool, a user must have a good understanding and working knowledge of the following:

- Electrical engineering and circuit design
- The ISO 26262 standard
- Documentation of the Library Compiler tool, such as the User Guide, at Library Compiler Online Help on SolvNet.
- This Functional Safety Manual
- The published list of safety-related defects for the Library Compiler tool available at Library Compiler Safety-Related Issues Master List. (https://solvnet.synopsys.com/retrieve/2808881.html)
- Applicability of the Library Compiler tool in the overall tool chain

Managing Known Safety-Related Defects

Synopsys maintains current information for every reported defect through STARs. The Library Compiler team evaluates each reported issue for potential impact on functional safety.

A list of all known safety-related defects for each release of Library Compiler is available on a SolvNet knowledge base article and is referenced from the *Library Compiler Release Notes*.

Library Compiler users must assess, as part of their own software tool criteria evaluation, the potential impact of the known safety-related defects in their design and must ensure mitigation of any relevant safety-related defects.

Managing New Releases

Synopsys can release new versions of the Library Compiler tool at any time to extend its functionality or to fix defects. When a new version is available, notification is posted on the SolvNet site. A subscription service is available for users to be notified of any new product releases.

When installing a new version of the Library Compiler tool, users must evaluate the impact of any known safety-related defects in their design by checking the accompanying *Library Compiler Release Notes* for the following:

- Any changes that apply to safety-related use cases
- List of known safety-related defects in the new version of the Library Compiler tool

In addition, users must refer to the latest version of this document, the *Library Compiler Functional Safety Manual*, available with the product release contents.

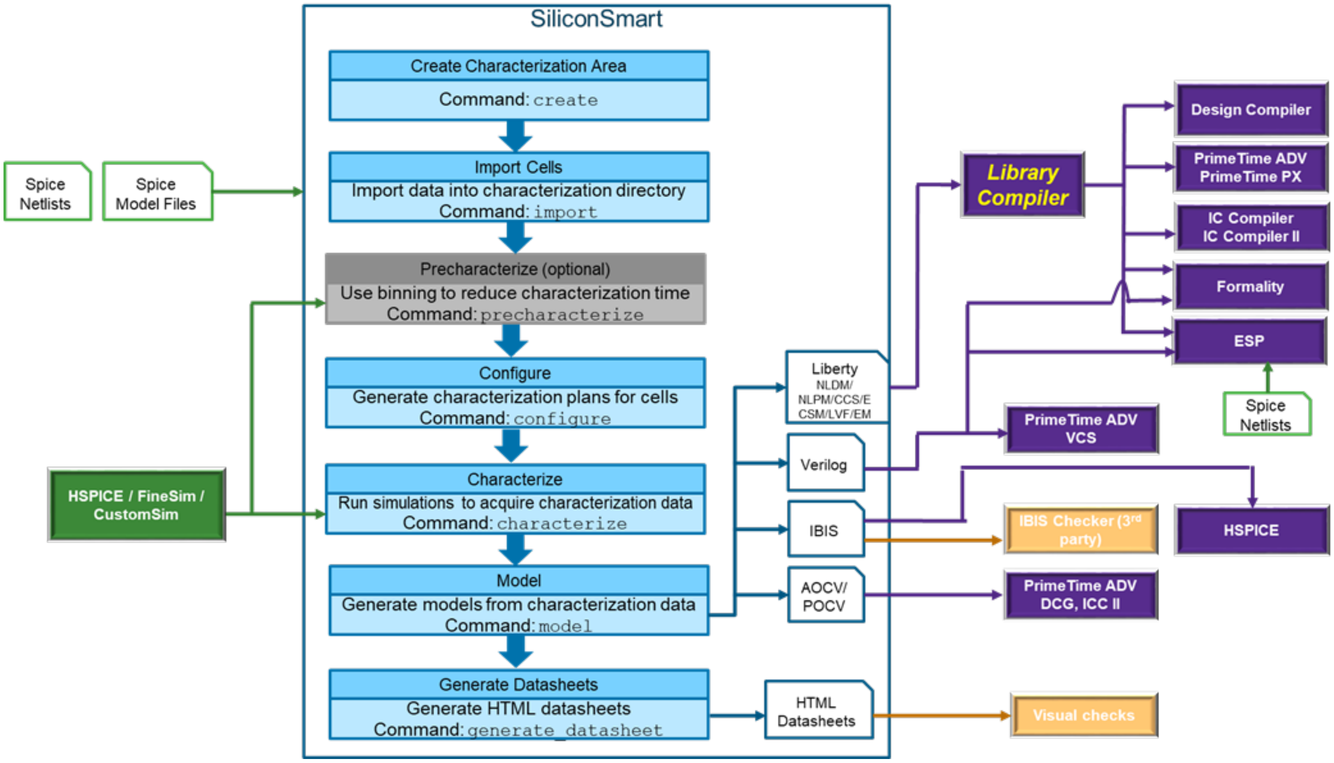
5
Synopsys Library Development Tool Chain

This section provides an overview of where the Library Compiler is used in the tool chain.

The ISO 26262 standard provides a methodology and requirements for software tool criteria evaluation and qualification (see ISO 26262-8, clause 11). It applies to software tools used for the development of safety-related designs where it is essential that the tool operates correctly without introducing or failing to detect errors in the safety-related design.

The suitability of a software tool to be used in the development of a safety-related design is determined in the software tool criteria evaluation, which results in a Tool Confidence Level (TCL): a level of confidence that the software tool does not introduce or fail to detect an error in the design without being noticed, and mitigated before the design is released as a safety-related product. This evaluation is best performed in the context of the overall software tool chain and development flow, in which the individual software tool is used. The following high-level diagram reflects the tool chain for which the Library Compiler tool is applicable.

Synopsys Library Development Tool Chain



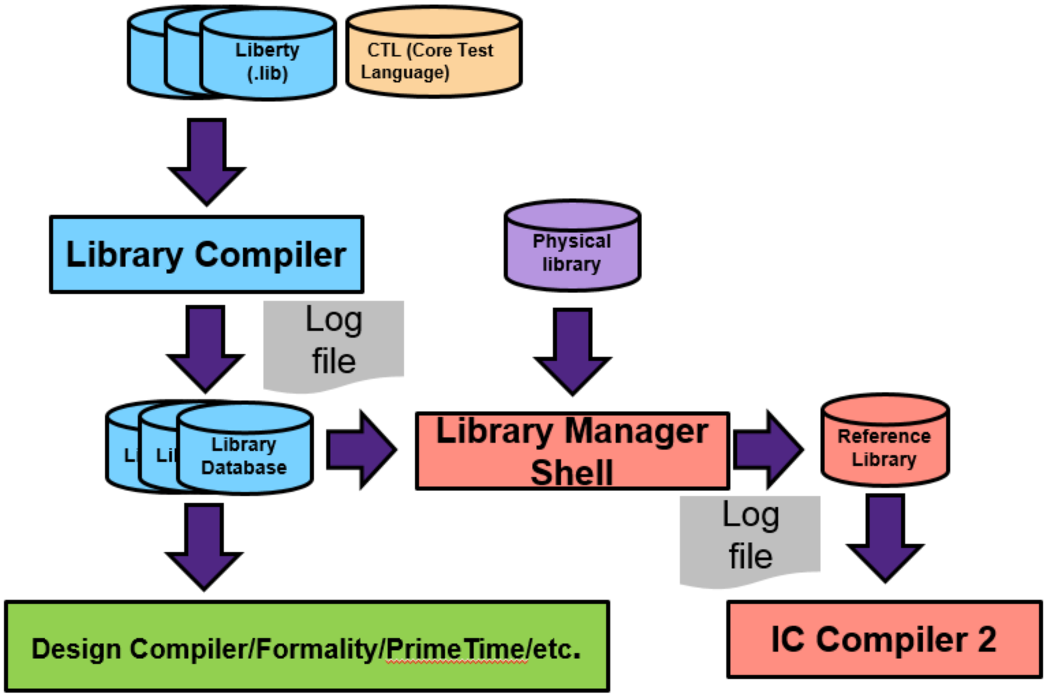
The Library Compiler Tool

- The main input is the Liberty logic library (.lib).

- Relevant Library Compiler commands are:
    - **read_lib**: checks the syntax for Liberty compliance and pre-processes data for consuming tools
    - **write_lib**: writes out library database for consuming tools
    - **report_lib**: reports library contents to users
    - **check_library**: checks the data content in libraries

- The Library Manager shell (library platform) helps to create reference library databases. The inputs to this tool are the physical and logic libraries. Other built-in checkers or screeners help to create reference library database for use with IC Compiler II for improved runtime.

- Starting with the version 2018.06, the Library Compiler tool will include the Library Manager shell.



6
Use Cases

This section describes the safety-qualified use cases of the Library Compiler tool. Users should also perform TCL determination based on their specific Use Cases.

Library Compiler Checks

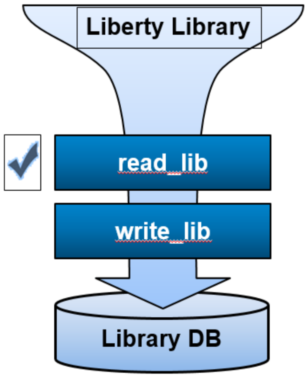
The Library Compiler tool is used to compile and validate an input liberty ASCII (in the .lib file format) library. The .lib file is thoroughly checked for accuracy and readiness for downstream optimization and analysis tools in the flow chain. The **read_lib** command along with the **write_lib** command creates a compiled library database (.db file) from the Liberty (.lib) input. The following steps are required to successfully create a .db file from a .lib file:

- The **read_lib** command is used to compile an input .lib into a .db file.
    - **Error** severity

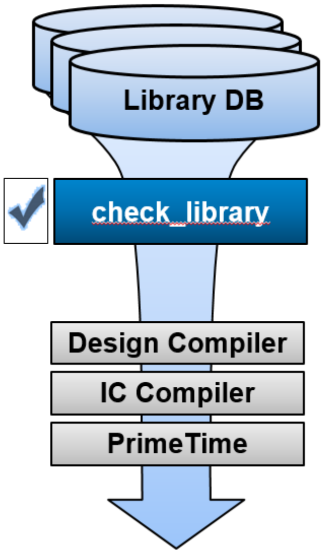
      Stops creation of the .db file requiring users to fix the source .lib for identified issue(s)
        - For example, an attribute is specified multiple times with different values
    - **Warning** severity

      Points to potential issues with the library model
        - User to proceed with caution
        - For example, syntax is specified but is unused by any cells
    - Information message

      Points to least severe data
        - For example, missing power information
- The **write_lib** command is used to write out compiled library databases for downstream Synopsys tools



- The Library database is then thoroughly checked for accuracy of readiness for downstream tools
  - The **check_library** command is used to qualify a library database with the following messages to the user:
    - **Error** severity is high and user must fix the issue
    - **Warning** severity is medium and the user must review the issue before proceeding
    - Information severity is low or a non-issue
- The **check_library** command also checks for flow readiness of library databases and physical libraries for downstream flow tools. The following are the important checks of the **check_library** command:
  - Multicorner-multimode (MCMM)
  - Unified Power Format (UPF)
  - Scaling
  - Advanced Waveform Propagation (AWP)



Use Case 1: Generation of Production-Quality Library Database

The goal of this use case is the generation of production-quality logic library database (.db) that is used as input for downstream tools such as Design Compiler, Formality, IC Compiler, VC LP, and PrimeTime.

To generate production-quality logic library databases, the Library Compiler tool is capable of performing checks for:

- Timing, timing variation, power, noise, functional and test views, power management functionality
- Electrical design rules including but not limited to electromigration, minimum and maximum capacitance, fanout, and input transitions
- Process, voltage and temperature (PVT) corners or operating conditions, wire-load models
- Verification of generated libraries, that is, library readiness check for different flows such as MCMM, UPF, scaling, AWP and consistency of different views

In this use case, the Library Compiler tool uses and generates the following main inputs and outputs.

- Inputs:
    - Liberty logic library (.lib)
    - Core test language (CTL)
    - Tcl command scripts
- Expected outputs:
    - Compiled library database (.db)
    - Compile run report logs
    - Optionally, users can report the contents of a library database using the **report_lib** and the Tcl commands

For this use case of the Library Compiler tool, the following conditions of use (constraints for the design and design environment, recommended procedures for the tool usage, etc.) shall be met:

- CoU-LC-001: User shall review all error and warning messages and take appropriate action.
- CoU-LC-002: User shall follow the *Library Compiler User Guide* (Chapter 3) and the *Library Compiler Quality Assurance System User Guide* (Chapters 1 and 2) for command usage.

- CoU-LC-003: User shall model libraries in the Liberty format as specified in the *Library Compiler User Guide* (Chapters 4 through 22) and the *Synopsys Logic Library Reference Manual* (Chapters 1 to 4).
- CoU-LC-004: User shall follow the *Library Compiler Quality Assurance System User Guide* (Chapter 3) and take appropriate action for validation of the libraries.

For this use case of the Library Compiler tool, the following assumptions of use (required actions to be taken by the tool user to prevent or detect design errors due to possible tool malfunctions) shall be met:

- AoU-LC-001: User shall review report and log files for errors and warnings and take appropriate action.
- AoU-LC-002: User shall review *Library Compiler User Guide* to follow the Liberty syntax in library modeling.
- AoU-LC-003: User shall review report file to check for the correct library input file and version number.
- AoU-LC-004: User shall check that all output files are generated with an up-to-date timestamp and are in the correct directory location.
- AoU-LC-005: User shall run **check_library** and **report_lib** commands to fully validate their compiled library.
- AoU-LC-007: User shall check log files of Design Compiler, IC Compiler/IC Compiler II, Formality and/or PrimeTime to ensure that .db file(s) have been read in successfully with no errors.

All analyzed failure modes and prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact Library Compiler tool version.

A software tool criteria evaluation performed by Synopsys according to ISO 26262-8, clause 11, which assumes the fulfillment of all conditions of use (CoU) and assumptions of use (AoU) as described above, results in a required tool confidence level:

TCL1 for *Library Compiler* Use Case 1: Generation of Production Quality Library Database

In this case, no further activities for software tool qualification are required.

Use Case 2: Generation of Reference Library for IC Compiler II

In this use case, the goal is to generate the reference library for IC Compiler II. The reference library (.clib file format) contains the logical, physical (full layout and the abstract views), and technology files in a compiled database.

In this use case, the Library Compiler tool uses and generates the following main inputs and outputs.

- Inputs:
  - Compiled Liberty database (.db)
  - Physical views (LEF, GDS or OASIS, CEL, FRAM)
  - Technology file
  - Tcl command scripts
- Expected outputs:
  - Reference library database (.clib)
  - Reports
  - Log Files

The physical view input files must contain syntax that adheres to their respective standards that can be referenced in the following documentation:

OASIS

http://ams.semi.org/ebusiness/standards/SEMIStandardDetail.aspx?ProductID=211&DownloadID=3889

http://ams.semi.org/ebusiness/standards/SEMIStandardDetail.aspx?ProductID=1948&DownloadID=3748

GDSII

http://bitsavers.informatik.uni-stuttgart.de/pdf/calma/GDS_II_Stream_Format_Manual_6.0_Feb87.pdf

LEF

http://www.ispd.cc/contests/18/lefdefref.pdf

For this use case of the Library Compiler tool, the following conditions of use (constraints for the design and design environment, recommended procedures for the tool usage, etc.) shall be met:

- CoU-LC-001: User shall review all error and warning messages and take appropriate action.

- CoU-LC-005: User shall follow the *ICC II Library Preparation User Guide* (Chapter 2) for proper command usage.
- CoU-LC-006: User shall follow *Synopsys Technology File and Routing Rules Reference Manual* (Chapters 1 through 3) to follow the syntax and physical library rules in library modeling.
- CoU-LC-007: User shall only use supported Tcl syntax as documented in the *Using Tcl With Synopsys Tools* user guide on SolvNet.
- CoU-LC-008: User shall follow the LEF, GDS and/or OASIS Standards Documentation for correct syntax and version support.

For this use case of the Library Compiler tool, the following assumptions of use (required actions to be taken by the tool user to prevent or detect design errors due to possible tool malfunctions) shall be met:

- AoU-LC-001: User shall review report and log files for errors and warnings and take appropriate action.
- AoU-LC-003: User shall review report file to check for the correct library input file and version number.
- AoU-LC-004: User shall check that all output files are generated with an up-to-date timestamp and are in the correct directory location.
- AoU-LC-006: User shall use the *ICC II Compiler Library Preparation User Guide* (Chapter 2) to validate the .clib.

All analyzed failure modes and prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact Library Compiler tool version.

A software tool criteria evaluation performed by Synopsys according to ISO 26262-8, clause 11, which assumes the fulfillment of all conditions of use (CoU) and assumptions of use (AoU) as described above, results in a required tool confidence level:

TCL1 for *Library Compiler* Use Case 2: Generation of Production Library for ICC II

In this case, no further activities for software tool qualification are required.

7
Limitations of Use Cases

This section describes all known limitations of the use cases mentioned in the previous section.

All known safety-related issues for the Library Compiler tool are listed in the Library Compiler Safety-Related Issues Master List available on SolvNet.

LIM-1: Do not use LCA features for production

Each release of the Library Compiler tool may contain hidden, undocumented features for testing or evaluation purposes, known as “Limited Customer Availability” (LCA) features. Use LCA features only for testing and evaluating the proposed new features, not for your production work.

Appendix A
Software Tool Information

This section provides general information about the Library Compiler software tool, which is needed by the tool user for performing his/her software tool criteria evaluation.

The following information about Library Compiler is required according to ISO 26262-8, for the planning of the usage of a software tool (clause 11.4.4) and the preparation of the own software tool criteria evaluation (clause 11.4.5).

Please note that some of the information below provided by Synopsys simply needs to be confirmed by the tool user and can be used without modification. Other information must be completed or updated by the tool user to reflect his/her actual situation.

Required Info	**Tool Information**	**Reference / Comment**
Tool vendor	Synopsys, Inc.	ISO 26262-8, 11.4.4.1.a
Tool name and version	Library Compiler	ISO 26262-8, 11.4.4.1.a To determine tool version, use: `report_version -options`
Tool use cases		ISO 26262-8, 11.4.4.1.c ISO 26262-8, 11.4.5.1.a To be completed by the tool user. Align with / verify against use cases described in Section 6 of this document.
Tool inputs and expected outputs		ISO 26262-8, 11.4.5.1.b To be completed by the tool user. Align with / verify against inputs and outputs described in Section 6 of this document.
Tool configuration and constraints		ISO 26262-8, 11.4.4.1.b ISO 26262-8, 11.4.5.1.c To be completed by the tool user. Align with / verify against CoU for the use cases described in Section 6 of this document.

Required Info	Tool Information	Reference / Comment
Tool environment (OS)	See Library Compiler Installation Notes at https://solvnet.synopsys.com/DownloadCenter. Click on **Library Compiler**, then the release number, such as N-2017.12 and then "View installation guide" for version-specific OS support.	ISO 26262-8, 11.4.4.1.d To be completed by the tool user. Align with / verify against the OS version evaluated by Synopsys. To determine Linux version, use: `uname -osr`
Tool environment (CAD tool chain)		ISO 26262-8, 11.4.4.1.d To be completed by the tool user. To determine name and version of your tool chain, please consult your CAD department.
Maximum ASIL	ASIL D	ISO 26262-8, 11.4.4.1.e
Tool qualification methods	Not applicable	ISO 26262-8, 11.4.4.1.f Software tool qualification is not required for Library Compiler
User manual and other usage guide documents	See Product Documentation and Support in Section 4 of this document. **OASIS** http://ams.semi.org/ebusiness/standards/SEMIStandardDetail.aspx?ProductID=211&DownloadID=3889 http://ams.semi.org/ebusiness/standards/SEMIStandardDetail.aspx?ProductID=1948&DownloadID=3748 **GDSII** http://bitsavers.informatik.uni-stuttgart.de/pdf/calma/GDS_II_Stream_Format_Manual_6.0_Feb87.pdf **LEF** http://www.ispd.cc/contests/18/lefdefref.pdf	ISO 26262-8, 11.4.4.2.a – d Tool user to include a link to these documents (Synopsys SolvNet or local copy), and to add any additional company-internal tool usage guidelines.

Required Info	Tool Information	Reference / Comment
Known software tool malfunctions, and appropriate work arounds ...	For limitations, refer to Section 7 of this document. https://solvnet.synopsys.com/retrieve/2808881.html	ISO 26262-8, 11.4.4.2.e Tool user to include a link to these documents (Synopsys SolvNet or local copy), and to add any additional company-internal work around descriptions.
Measures for the detection of tool malfunctions ...		ISO 26262-8, 11.4.4.2.f To be completed by the tool user. Align with / verify against AoU for the use cases described in Section 6 of this document.
Measures for the detection of tool malfunctions ...		ISO 26262-8, 11.4.4.2.f To be completed by the tool user. Align with / verify against AoU for the use cases described in Section 6 of this document.

Appendix B
Complete List of CoU and AoU IDs

The complete list of Conditions of Use (CoU) for Library Compiler is in the table below. CoU defines a condition of the design, software tool, design environment, or situation that is assumed and required to be fulfilled by the user.

ID	**Description**
CoU-LC-001	User shall review all error and warning messages and take appropriate action.
CoU-LC-002	User shall follow the *Library Compiler User Guide* (Chapter 3) and the *Library Quality Assurance System User Guide* (Chapters 1 and 2) for command usage.
CoU-LC-003	User shall model libraries in Liberty format as specified in the *Library Compiler User Guide* (Chapters 4 through 22) and the *Synopsys Logic Library Reference Manual* (Chapters 1 to 4).
CoU-LC-004	User shall follow the *Library Quality Assurance System User Guide* (Chapter 3) and take appropriate action for validation of the libraries.
CoU-LC-005	User shall follow the *ICC II Library Preparation User Guide* (Chapter 2) for proper command usage.
CoU-LC-006	User shall follow *Synopsys Technology File and Routing Rules Reference Manual* (Chapters 1 through 3) to follow the syntax and physical library rules in library modeling.
CoU-LC-007	User shall only use supported Tcl syntax as documented in *Using Tcl With Synopsys Tools* user guide on Solvnet.
CoU-LC-008	User shall follow the LEF, GDS and/or OASIS Standards Documentation for correct syntax and version support.

The complete list of Assumptions of Use (AoU) for Library Compiler is in the table below. AoU defines an action that is assumed and required to be taken by the user of a software tool.

ID	**Description**
AoU-LC-001	User shall review report and log files for errors and warnings and take appropriate action.
AoU-LC-002	User shall review *Library Compiler User Guide* to follow the Liberty syntax in library modeling.

ID	**Description**
AoU-LC-003	User shall review report file to check for the correct library input file and version number.
AoU-LC-004	User shall check that all output files are generated with an up-to-date timestamp and are in the correct directory location.
AoU-LC-005	User shall run **check_library** and **report_lib** commands to fully validate their compiled library.
AoU-LC-006	User shall use the *ICC II Compiler Library Preparation User Guide* (Chapter 2) to validate the .clib.
AoU-LC-007	User shall check log files of Design Compiler, IC Compiler/IC Compiler II, Formality and/or PrimeTime to ensure that .db file(s) have been read in successfully with no errors.