

# **ZeBu<sup>®</sup> Server 4 Release Notes**

---

Version V-2024.03-1, July 2024



# Copyright and Proprietary Information Notice

© 2024 Synopsys, Inc. This Synopsys software and all associated documentation are proprietary to Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, modification, or distribution of the Synopsys software or the associated documentation is strictly prohibited.

## 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

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at <https://www.synopsys.com/company/legal/trademarks-brands.html>. All other product or company names may be trademarks of their respective owners.

## Free and Open-Source Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

## 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.

[www.synopsys.com](http://www.synopsys.com)

# Contents

---

About This Book .....	8
Contents of This Book .....	8
Related Documentation .....	9
Typographical Conventions .....	10

---

<b>1. Installation .....</b>	<b>12</b>
Operating Systems .....	12
Installation Procedure .....	13
Licenses .....	14
ZeBu Software .....	14
Xilinx Place-and-Route Software for ZeBu Server 4 .....	15
License Features for Unified Compile .....	15
License Features for Debug Products .....	15
License Feature for ZeBu Simulation Acceleration .....	16
ZeBu PCIe Driver and Firmware .....	16
Latest Versions of the PCIe Drivers .....	16
Updating the PCIe Firmware .....	16
Regenerating the Setup Directory (\$ZEBU_SYSTEM_DIR) .....	16
Hardware Initialization .....	17
Recompiling the Design .....	17

---

<b>2. Modified Default Settings .....</b>	<b>18</b>
Modified Default Settings in V-2024.03-1 .....	18
Compile Time and Stability Settings .....	18
Common Elaboration Flow in Verdi (Vega2) .....	18
Runtime Default Settings .....	19
Reporting SVA Failures .....	19
Unified Compile Command Updates .....	19
Deprecated Commands .....	19
New Commands .....	21
Modified Default Settings in V-2024.03 .....	22

## Contents

Compile Time and Stability Settings . . . . .	22
Support for Optimizing Uninitialized Registers . . . . .	22
Default Limit for Maximum Loop Iterations Updated . . . . .	22
Deterministic BRAM Dual Port Access . . . . .	23
Optimize Unconnected or Constant Ports . . . . .	23
Virtual File System (VFS) . . . . .	23
URAM Capacity Improvement by Address Folding . . . . .	23
Enhanced Verilog Force Release . . . . .	23
Stimuli Capture and Replay and Waveform Expansion are Independent Features . . . . .	23
Hydra Architecture Update . . . . .	24
Performance Default Settings . . . . .	25
CDP Lite is Automatically Enabled . . . . .	25
Designs Using \$time are Automatically Optimized . . . . .	25
MCP Information to Relax Feedback Paths . . . . .	25
MCP Information to Relax Filter-Barrier Paths . . . . .	25
Default Transactor Fetch Mode Feature Support . . . . .	25
Drop Asynchronous <code>set/reset</code> Clock Domain . . . . .	26
Unified Compile Command Updates . . . . .	26
Obsolete Commands . . . . .	26
Deprecated Commands . . . . .	27
New Commands . . . . .	29
<b>3. New Features and Enhancements . . . . .</b>	<b>31</b>
New Features and Enhancements in V-2024.03-1 . . . . .	31
Compilation Stability Improvements . . . . .	31
Consistent Naming Style for zPar Metric String IDs . . . . .	31
Improvements to zAudit Reporting . . . . .	32
Runtime Improvements . . . . .	32
Automatically Capturing Streaming Data at Runtime . . . . .	33
Analog Mixed Signal Support . . . . .	33
Compress XML File Generation During Functional Coverage . . . . .	33
\$callback Support . . . . .	33
C++ API for Erasing Memories . . . . .	34
Enable Periodic Save for Functional Coverage . . . . .	34
Fault Emulation . . . . .	34
FSDB Stitch Mode Support in zSimzilla . . . . .	35
Memory Word Replication Using Firmware . . . . .	35
Relaunching Spawned Jobs Using zSimzilla . . . . .	36
Reset Support in CEL . . . . .	36
zsvaReport Enhancement . . . . .	37
Performance Improvements . . . . .	37

## Contents

Support for Variable Delay in RTL Clock Replication . . . . .	37
Reshaping of User Memories . . . . .	38
Debug Capabilities . . . . .	38
Support for VHDL Modules in Toggle Coverage . . . . .	38
New Features and Enhancements in V-2024.03 . . . . .	38
Compilation Stability Improvements . . . . .	38
Error Displayed When Unified XMR Flow is Disabled . . . . .	38
Runtime Improvements . . . . .	39
Identifiers Added for Runtime Warnings . . . . .	39
Checkpoint and Restart Features . . . . .	39
Sample Control in zRci . . . . .	40
zTune: Printing Hardware Information . . . . .	40
zTune: New Markers for Enhanced Performance . . . . .	40
zdpiReport Enhancement . . . . .	40
zRci Options . . . . .	40
Debug Capabilities . . . . .	41
Vega2 is Enabled for the Verdi Flow . . . . .	41
zSimzilla Enhanced to Improve Faster Waveform Conversion . . . . .	41
New C++ Classes and APIs . . . . .	42
<hr/>	
<b>4. LCA Features . . . . .</b>	<b>44</b>
LCA Features in V-2024.03-1 . . . . .	44
ECO Flow for Dynamic Triggers . . . . .	44
ZRM Performance Mode is Enabled by Default . . . . .	44
LCA Features in V-2024.03 . . . . .	45
Checkpoint and Restart Using DMTCP . . . . .	45
Toggle Coverage . . . . .	45
<hr/>	
<b>5. Deprecated Features . . . . .</b>	<b>46</b>
Deprecated Features in V-2024.03-1 . . . . .	46
Deprecated Databases . . . . .	46
Deprecated Features in V-2024.03 . . . . .	46
Hardware Updates . . . . .	46
C++ APIs and Classes . . . . .	47
<hr/>	
<b>6. Obsolete Features . . . . .</b>	<b>48</b>
Obsolete Features in V-2024.03 . . . . .	48

Obsolete APIs .....	48
<hr/>	
<b>7. Limitations .....</b>	<b>50</b>
Operating Systems .....	50
Compilation Interface (zCui) .....	50
zCui uses /bin/sh and ignores .cshrc Aliases or Functions .....	50
Limitations in the Compilation View .....	50
Compilation .....	51
Unsupported Performance Oriented Partitioning with Direct ICE .....	51
Unsupported Timing Optimization Features .....	51
Memory Modeling .....	51
Limitation on Read/Write Priority for Multi-port <code>zrm</code> Memories .....	51
Automatic Memory Loading .....	52
Runtime .....	52
Initialization Order in a C++/C Testbench .....	52
Limitation on Fast Hardware State .....	53
Sampling Clocks for Waveform Capture .....	53
Disk Resource Conflicts for ZTDB Files .....	54
Memory Initialization not Carried out in Some C/C++ Testbenches .....	54
Clock Declaration in the designFeatures File .....	55
Specification of the Initial Clock State .....	55
Message Size in Clock-Delay Mode .....	55
SystemVerilog Assertions (SVAs) .....	56
Unsupported SVA Features .....	56
Full Report Mode .....	56
Report only Failure Mode .....	56
zPostRunDebug .....	56
Limitation on Primary Clocks .....	56
Limitation with Direct-ICE and SMART Z-ICE Clocks .....	57
ZEMI-3 .....	57
<hr/>	
<b>8. Fixed STAR List .....</b>	<b>58</b>
<hr/>	
<b>9. Known Issues .....</b>	<b>62</b>
<hr/>	
<b>10. Version Compatibility .....</b>	<b>63</b>

## Contents

Operating System Compatibility . . . . .	63
Hardware Compatibility . . . . .	63
Checking the Need for a New Xilinx License File . . . . .	63
Xilinx Vivado License for ZeBu Server 4 Hardware . . . . .	63
New Version of xilinuxd License Daemon for ZeBu Server 4 Hardware . . . . .	64
Synopsys Interoperable Technologies . . . . .	65
Third-Party Tools Compatibility . . . . .	65
GCC Version and Recommendations . . . . .	66
<hr/>	
<b>11. Documentation . . . . .</b>	<b>67</b>
Accessing ZeBu Server 4 Documentation in SolvNetPlus . . . . .	67
Logging Into SolvNetPlus . . . . .	67
Using Search . . . . .	68
Viewing the ZeBu Server 4 Documentation Page . . . . .	69
Downloading the ZeBu Server 4 Documentation Package . . . . .	70
Documentation in Installation Area . . . . .	71
Accessing ZeBu Documentation Using SmartSearch . . . . .	71
Accessing ZeBu Documentation in HTML Format . . . . .	72
Accessing ZeBu Documentation in PDF . . . . .	74
Viewing PDF Files . . . . .	74
Searching in All PDFs . . . . .	75
Additional Navigation in ZeBu Documentation . . . . .	76

# Preface

---

This chapter has the following sections:

- [About This Book](#)
  - [Contents of This Book](#)
  - [Related Documentation](#)
  - [Typographical Conventions](#)
- 

## About This Book

The *ZeBu® Server 4 Release Notes* describes new features available in the V-2024.03-1 release of the ZeBu software.

The present software version is applicable to ZeBu Server 4.

This document is intended for users who are familiar with the ZeBu product range.

---

## Contents of This Book

The *ZeBu® Server 4 Release Notes* document has the following chapters:

Chapter	Description
<a href="#">Installation</a>	Provides a list of required components to install this release
<a href="#">Modified Default Settings</a>	Provides the changes to default settings to improve the compilation stability
<a href="#">New Features and Enhancements</a>	Provides a brief description about new features or any changes to existing feature
<a href="#">LCA Features</a>	Provides a list of LCA features available in this release
<a href="#">Deprecated Features</a>	Provides a list of features deprecated with this release
<a href="#">Obsolete Features</a>	Provides a list of features that are marked obsolete with this release



Chapter	Description
<a href="#">Limitations</a>	Provides a brief description of limitations in this release
<a href="#">Fixed STAR List</a>	Provides a list of STARs fixed in this release
<a href="#">Known Issues</a>	Provides information on known issues in this release
<a href="#">Version Compatibility</a>	Provides information on different tools compatibility with this release
<a href="#">Documentation</a>	Provides a list of enhancements for documentation

## Related Documentation

Document Name	Description
ZeBu Getting Started Guide	Provides brief information about Synopsys' emulation system - ZeBu.
ZeBu Server 4 Site Planning Guide	Describes planning for ZeBu Server 4 hardware installation.
ZeBu Site Administration Guide	Provides information on administration tasks for ZeBu Server 4 hardware. It includes software installation.
ZeBu Power Estimation User Guide	Provides the power estimation flow and the tools required to estimate the power on a System on a Chip (SoC) in emulation.
ZeBu Server 4 Smart Z-ICE Interface User Guide	Provides physical description of the Smart Z-ICE interface and the steps to instantiate and use it on ZeBu Server 4.
ZeBu Server 4 Release Notes	Provides enhancements and limitations for new ZeBu Server 4 releases.

Document Name	Description
<i>ZeBu Debug Guide</i>	Provides information on tools you can use for debugging.
<i>ZeBu User Guide</i>	Provides detailed information on using ZeBu.
<i>ZeBu Debug Methodology Guide</i>	Provides debug methodologies that you can use for debugging.

Document Name	Description
<i>ZeBu Unified Command-Line User Guide</i>	Provides the usage of Unified Command-Line Interface (UCLI) for debugging your design.
ZeBu UTF Reference Guide	Describes Unified Tcl Format (UTF) commands used with ZeBu.
ZeBu Power Aware Verification User Guide	Describes how to use Power Aware verification in ZeBu environment, from the source files to runtime.
<i>ZeBu Functional Coverage User Guide</i>	Describes collecting functional coverage in emulation.
Simulation Acceleration User Guide	Provides information on how to use Simulation Acceleration to enable cosimulating SystemVerilog testbenches with the DUT
ZeBu Verdi Integration Guide	Provides Verdi features that you can use with ZeBu. This document is available in the Verdi documentation set.
ZeBu Runtime Performance Analysis With zTune User Guide	Provides information about runtime emulation performance analysis with zTune.
ZeBu Custom DPI Based Transactors User Guide	Describes ZEMI-3 that enables writing transactors for functional testing of a design.
<i>ZeBu LCA Features Guide</i>	Provides a list of Limited Customer Availability (LCA) features available with ZeBu.
ZeBu Transactors Compilation Application Note	Provides detailed steps to instantiate and compile a ZeBu transactor.
ZeBu zManualPartitioner Application Note	Describes the zManualPartitioner feature for ZeBu. It is a graphical interface to manually partition a design.
ZeBu Hybrid Emulation Application Note	Provides an overview of the hybrid emulation solution and its components.

## Typographical Conventions

This document uses the following typographical conventions:

To indicate	Convention Used
Program code	<i>OUT</i> <= <i>IN</i> ;
Object names	<i>OUT</i>
Variables representing objects names	< <i>sig-name</i> >

To indicate	Convention Used
Message	Active low signal name '<sig-name>' must end with _X.
Message location	<i>OUT</i> <= IN;
Reworked example with message removed	<i>OUT_X</i> <= IN;
Important Information	<b>NOTE:</b> This rule...

The following table describes the syntax used in this document:

Syntax	Description
[ ] (Square brackets)	An optional entry
{ } (Curly braces)	An entry that can be specified once or multiple times
(Vertical bar)	A list of choices out of which you can choose one
... (Horizontal ellipsis)	Other options that you can specify

# 1

## Installation

---

This chapter provides information on the enhancements made to the installation-related tasks and features.

See the following sections:

- [Operating Systems](#)
- [Installation Procedure](#)
- [Licenses](#)
- [ZeBu PCIe Driver and Firmware](#)
- [Regenerating the Setup Directory \(\\$ZEBU\\_SYSTEM\\_DIR\)](#)
- [Hardware Initialization](#)
- [Recompiling the Design](#)

---

## Operating Systems

The ZeBu V-2024.03-1 release supports the following QSC-V operating systems (OS):

- Red Hat Enterprise Linux 7.3+
- Red Hat Enterprise Linux 8+
- CentOS 7.3.1611+
- SUSE Linux Enterprise Server 12-SP5+
- SUSE Linux Enterprise Server 15-SP2+
- Rocky Linux 8.4+
- AlmaLinux: 8.4+

For details about supported OS versions across releases, see [Compute Platforms Roadmap](#).

The compilation software is expected to work on any of the OS versions listed earlier. However, runtime of the ZeBu V-2024.03-1 release was only tested using the following OS versions:

- Red Hat Enterprise Linux 8.6
- Rocky Linux 8.6
- SUSE Linux Enterprise Server 15.3
- AlmaLinux 8.6
- CentOS Linux release 7.3

Therefore, it is recommended to use these OS versions (or the RHEL equivalent to CentOS) for runtime.

---

## Installation Procedure

To install the ZeBu software, you must download the Synopsys Installer version 5.7 and the ZeBu software package from SolvNetPlus.

For more information, see *ZeBu Site Administration Guide* based on your hardware.

### Note:

Before launching the installation, ensure that the `DISPLAY` variable is correctly set in your installation environment.

Observe the layout in the `$ZEBU_ROOT` and the installation directory:

- `$ZEBU_ROOT`
  - Hardware-specific child directory: `$ZEBU_ROOT/{zs4,zs5,ep1,ep2}`
  - Hardware-specific files in `$ZEBU_ROOT/etc` transformed to links pointing inside these directories
- Installation directory (`ZEBU_ROOT`)
  - Hardware-specific sub directories
  - Hardware-specific files/directories in `$ZEBU_ROOT/etc` is replaced by links to files/directories in `$ZEBU_ROOT/{zs4,zs5,ep1,ep2}`

---

## Licenses

This section provides information about the list of licenses that you must procure or renew to use the existing or new features of ZeBu.

- [ZeBu Software](#)
- [Xilinx Place-and-Route Software for ZeBu Server 4](#)
- [License Features for Unified Compile](#)
- [License Features for Debug Products](#)
- [License Feature for ZeBu Simulation Acceleration](#)

---

### ZeBu Software

ZeBu version V-2024.03 onwards requires new license files for the `snpslmd` license daemon.

To obtain the latest license key file for your site, perform the following steps:

1. Navigate to the [SolvNetPlus Downloads Centre - Synopsys](#) website. The **SolvNetPlus Downloads** page is displayed.
2. From the displayed list of products on the **SolvNetPlus Downloads** page, scroll and choose **Synopsys Common Licensing**.

**Note:**

If you are a Linux user, you need to additionally download the **Synopsys Installer 5.7**.

3. Before you choose a version from the displayed list, see the [Synopsys Licensing QuickStart Guide](#) for details.

If an out-of-date license daemon is used, the following error message is displayed:

```
[SCL] Error: FLEXlm version of the application is higher than the license
daemon version. Please upgrade the license daemon to latest version
(SCL-601)
```

If an out-of-date license key file is used, the following error message is displayed:

```
License server system does not support this version of this feature.
```

---

## Xilinx Place-and-Route Software for ZeBu Server 4

For Xilinx Vivado software, the shipped package is now 2022.1 for which the Xilinx Version Limit date is 2022.04. You should check your Vivado license file as described in the Xilinx Vivado License for ZeBu Server 4 Hardware section.

If the Xilinx Version Limit of your Vivado license file is not 2022.04 or later, you must request a new Vivado license file from Synopsys.

---

## License Features for Unified Compile

To use the Unified Compile feature, download and install the supported releases of the following Synopsys products:

- VCS MX
- VCS Native Low Power (NLP)
- VC LP Static Low-Power
- SpyGlass

The license features required by these Synopsys products are included in the new license files downloaded from SmartKeys as mentioned in section [ZeBu Software](#).

Some ZeBu Server part numbers are for specific customers and do not include the VCS, VCS Native Low Power (NLP), VC LP Static Low-Power, and SpyGlass compilation features with their ZeBu Server license files. These customers are required to use VCS, VCS Native Low Power (NLP), VC LP Static Low-Power, and SpyGlass licenses acquired independently.

---

## License Features for Debug Products

To benefit from enhanced debugging, download and install the supported releases of these additional Synopsys products:

- Verdi

The license features required by these Synopsys products are included in the new license files downloaded from SmartKeys as mentioned in section [ZeBu Software](#).

Some ZeBu Server part numbers do not include the Verdi license features with their ZeBu Server license files. In such cases, you must use Verdi license acquired independently for debugging.

---

## License Feature for ZeBu Simulation Acceleration

For ZeBu Simulation Acceleration, use VCS licenses acquired independently for runtime purposes.

---

## ZeBu PCIe Driver and Firmware

ZeBu Server 4 V-2024.03-1 software requires a specific PCIe driver to connect to ZeBu Server as described in the following sections:

- [Latest Versions of the PCIe Drivers](#)
  - [Updating the PCIe Firmware](#)
- 

### Latest Versions of the PCIe Drivers

The minimum version of the PCIe driver for ZeBu Server 4 is version 5. However, it is recommended to use the latest version, which is version 8.

The instructions to download the latest ZeBu Device Drivers package is available in the following article on SolvNetPlus: [Instructions to Download the ZeBu Device Drivers Package for ZeBu Server 4](#)

---

### Updating the PCIe Firmware

To work with the new PCIe driver, update the firmware of the PCIe board using the `zUpdate` utility.

For more information, see the **Updating PCIe Boards Firmware** section in the *ZeBu Site Administration Guide*.

---

## Regenerating the Setup Directory (\$ZEBU\_SYSTEM\_DIR)

To upgrade your ZeBu system to run with the V-2024.03-1 release, perform the following steps:

1. Launch `zSetupSystem` to generate a new setup directory.
2. Set the `$ZEBU_SYSTEM_DIR` variable to the newly generated setup directory.



---

## Hardware Initialization

Before proceeding with emulation runtime on V-2024.03-1 version, you must initialize your ZeBu system for V-2024.03-1 with `zUtils -initSystem`.

**Note:**

You must always use `zUtils -initSystem` from the latest ZeBu software version.

---

## Recompiling the Design

Before proceeding with emulation runtime using V-2024.03-1, you must recompile your design with V-2024.03-1.

If you attempt to launch emulation runtime with V-2024.03-1 version before recompiling your design, the following error message is displayed:

```
### internal error [ZDB1297E] : Could not read children
### internal error [ZDB1297E] : zrdb_ZTreeDB.cc, line 1733 :
  deserialize_zdriverthing
### fatal error in DATABASE [ZDB1276F] : Error occured while loading
  database. Check the compilation consistency
-- ZeBu : testbench : ERROR : ZPRIV1629E : Error while loading the
  database:
-- ZeBu : testbench : ERROR : Error occured while loading database. Check
  the compilation consistency
-- ZeBu : testbench : ERROR : ZPRIV2454E : An error occurred while
  loading the runtime database (see above for more information).
```

# 2

## Modified Default Settings

---

This chapter describes the changes in default settings that have been done to improve the compilation stability and performance.

For more information, see the following sections:

- [Modified Default Settings in V-2024.03-1](#)
- [Modified Default Settings in V-2024.03](#)

---

### Modified Default Settings in V-2024.03-1

The following changes are default in V-2024.03-1:

- [Compile Time and Stability Settings](#)
- [Runtime Default Settings](#)
- [Unified Compile Command Updates](#)

---

### Compile Time and Stability Settings

The default settings to improve compile time and stability are as follows:

- [Common Elaboration Flow in Verdi \(Vega2\)](#)

### Common Elaboration Flow in Verdi (Vega2)

Starting with the V-2024.03-1 release, the common elaboration flow in Verdi (Vega2) is enabled by default for ZeBu.

Regenerate any precompiled libraries by adding the following to the Analysis commands:

```
-kdb=common_elab
```

To fall back to the old flow, use the following UTF command:

```
debug -verdi_db old
```

---

## Runtime Default Settings

The default runtime settings are as follows:

- [Reporting SVA Failures](#)

## Reporting SVA Failures

Starting with V-2024.03-1, SystemVerilog Assertion (SVA) failures are reported based on the default clock during Stimuli Replay.

For details, see the *ZeBu Debug Guide*.

---

## Unified Compile Command Updates

The Unified Compile command updates are as follows:

- [Deprecated Commands](#)
- [New Commands](#)

## Deprecated Commands

The following commands or command options are deprecated from V-2024.03-1 release. These commands return a warning or an error message when detected in the UTF file. They are either replaced by other commands or are related to new defaults:

- [debug Command Options](#)
- [fusa](#)
- [probe\\_signals Options](#)
- [zgate Command Options](#)
- [zpar Command Options](#)

### debug Command Options

Starting with V-2024.03-1, the following `debug` command options are now removed and using them results in an error:

- `-csa <bool>`
- `-csa_header <bool>`
- `-enable_hwtop_ports_access <none|input|output|inout|all|select|speed_zprd>`

- `-set_dyn_probes_on_dut_ios <bool>`
- `-set_dyn_probes_on_memory_ios <bool>`

### **fusa**

Starting with V-2024.03-1, the `fusa` command is deprecated.

Use the `fault_emulation` command instead.

### **probe\_signals Options**

Starting with V-2024.03-1, the following error message is displayed if the `probe_signals` command options are used:

```
Error-[UTF_UUCO] Unknown UTF command option <project.utf>, <line number>.
The specified option '-size_gtr_than' to UTF command is unknown.
```

ZeBu compile fails and you are required to correct the UTF file to remove the unknown command options.

For information about the command options that were deprecated in V-2024.03, see `probe_signals` Command Options in the [Deprecated Commands](#) section.

### **zgate Command Options**

Starting with V-2024.03-1, the following `zgate` command options are now removed and using them results in an error:

- `-depth <int>`
- `-access [all|seq|ports]`

### **zpar Command Options**

Starting with V-2024.03-1, the following `zpar -advanced_command` command options are deprecated and using them results in an error:

```
zpar -advanced_command {System pinAssignmentEffort low}
zpar -advanced_command {System pinAssignmentEffort medium}
zpar -advanced_command {System pinAssignmentEffort high}
```

### **Note:**

It is not recommended to use these commands because the default `zPar` setting (`zap`) is optimal in terms of compile-time and in achieving `driverClk` frequency.

## New Commands

The following new commands are introduced in this release:

- [clock\\_frequency](#)
- [fault\\_emulation](#)
- [read\\_classification](#)

### clock\_frequency

A new UTF command `clock_frequency` is supported from V-2024.03-1 release.

```
clock_frequency [-file <string>]
```

This command enables clock frequencies support.

For more information about this command, refer to the *ZeBu® Unified Tcl Format Reference Guide*.

### fault\_emulation

A new UTF command `fault_emulation` is supported from V-2024.03-1 release. This command replaces the `fusa` command which is now deprecated.

```
fault_emulation [-campaign <string>] [-dut_path <string>] [-fdb_name  
<string>] [-fdb_server <string>] [-object_not_found <string>]
```

This command indicates the fault emulation database and the campaign name.

For more information about this command, refer to the *ZeBu® Unified Tcl Format Reference Guide*.

### read\_classification

A new UTF command `read_classification` is supported from V-2024.03-1 release.

```
read_classification <classifFileNoOpt> [-edif <bool>] [-file <file>]
```

This command invokes the classification reader.

For more information about this command, refer to the *ZeBu® Unified Tcl Format Reference Guide*.

---

## Modified Default Settings in V-2024.03

The following changes are default in V-2024.03:

- [Compile Time and Stability Settings](#)
- [Performance Default Settings](#)
- [Unified Compile Command Updates](#)

---

### Compile Time and Stability Settings

The default settings to improve compile time and stability are as follows:

- [Support for Optimizing Uninitialized Registers](#)
- [Default Limit for Maximum Loop Iterations Updated](#)
- [Deterministic BRAM Dual Port Access](#)
- [Optimize Unconnected or Constant Ports](#)
- [Virtual File System \(VFS\)](#)
- [URAM Capacity Improvement by Address Folding](#)
- [Enhanced Verilog Force Release](#)
- [Stimuli Capture and Replay and Waveform Expansion are Independent Features](#)
- [Hydra Architecture Update](#)

### Support for Optimizing Uninitialized Registers

Starting with the V-2024.03 release, the Synthesizer tags the uninitialized registers with the `NOINIT` attribute for optimization by default.

To restore the previous behavior, use the following UTF command:

```
Synthesis -advanced_command{Compile:UseExplicitInits=false}
```

### Default Limit for Maximum Loop Iterations Updated

Starting with the V-2024.03 release, the default limit of maximum loop iterations is updated from 16384 to 35000.

The default limit can be overridden using the following UTF command: `synthesis -max_loop_iterations <int>`.

## Deterministic BRAM Dual Port Access

Starting with the V-2024.03 release, BRAM dual port protection is enabled by default.

To rollback to the previous behavior, use the following command:

```
memory_preferences -manage_dual_port_bram_access false
```

## Optimize Unconnected or Constant Ports

Starting with V-2024.03, all unconnected and constant memory ports are made inactive by default to optimize memory usage.

To restore the previous behavior, run the following UTF command:

```
ztopbuild -advanced_command {memory_opt port_opt -disable}
```

## Virtual File System (VFS)

Starting with V-2024.03, Virtual File System (VFS) is enabled by default to reduce stress on the network/disks.

To restore the previous behavior, use the UTF command:

```
synthesis -use_vfs false
```

## URAM Capacity Improvement by Address Folding

Starting with V-2024.03, memory capacity for URAM mapped memories can be optimized. URAM can be folded to optimize usage depending on the design memory shape.

To restore the previous behavior, use the following UTF command:

```
memory_preferences -advanced_command { config_uram -enable_folding  
false }
```

## Enhanced Verilog Force Release

Starting with V-2024.03, LRM compliant Verilog force release is enabled by default.

This implementation is handled at synthesis level to be more accurate according to LRM.

To restore the legacy flow, use the following UTF command:

```
verilog_force_release -enable_wls false
```

## Stimuli Capture and Replay and Waveform Expansion are Independent Features

In the earlier releases, when the Transactional Stimuli Capture and Replay feature was enabled using the `debug -offline_debug true` UTF command, waveform expansion was automatically enabled.

Starting with V-2024.03, these two features are automatically enabled and are independent of each other.

To disable Stimuli Capture and Replay, use the following UTF command:

```
debug -offline_debug false
```

To disable waveform expansion, use the following UTF command:

```
debug -waveform_reconstruction false
```

## Hydra Architecture Update

Hydra architecture was enabled by default starting with the S-2021.09 release.

With the current V-2024.03 release, the following additional updates are made default:

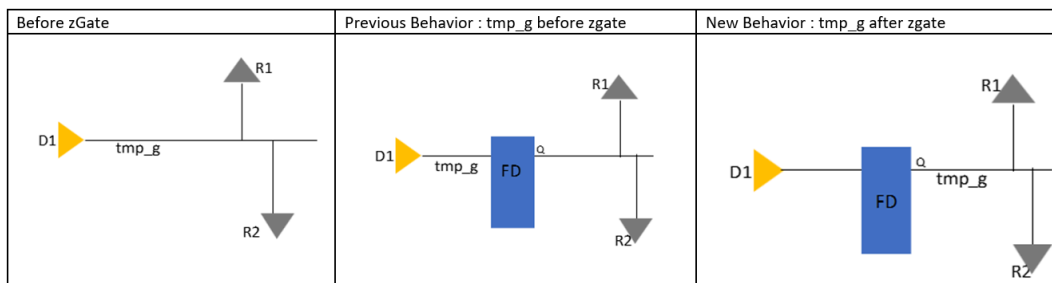
- [zgate Impact on Waveform](#)
- [Support for Waveform Output With Multidriviers](#)

### zgate Impact on Waveform

In V-2024.03, zgates are applied as driver of the specified signal. This can lead to shift in waveform for such signals compared to waveform from earlier releases. Before, `zgate` was connected as the load of the specified signal.

Consider the following example where `tmp_g` is declared as a wire in the RTL file. Note the three states displayed in the figure after the following UTF command was run:

```
zgate fd -clock_name hw_top.i_dut.clk -rtlname  
hw_top.i_dut.i_instrumented.tmp_g.
```



You observe a shift in waveforms on this signal because the signal on which a `zgate` is applied is now at the output of the `zgate` and no more at the input (`zgate` effect).

For information about UTF commands deprecated in hydra, see [Deprecated Commands](#).

### Support for Waveform Output With Multidriviers

In earlier ZeBu releases, additional probes were automatically added at the output of the multidriviers without any user commands. In V-2024.03, with the updated architecture,



these additional probes are no longer needed and are removed. This may have an impact on the waveforms of the output of the multidrivers.

---

## Performance Default Settings

The default settings to improve performance are as follows:

- [CDP Lite is Automatically Enabled](#)
- [Designs Using \\$time are Automatically Optimized](#)
- [MCP Information to Relax Feedback Paths](#)
- [MCP Information to Relax Filter-Barrier Paths](#)
- [Default Transactor Fetch Mode Feature Support](#)
- [Drop Asynchronous set/reset Clock Domain](#)

### CDP Lite is Automatically Enabled

Starting with the V-2024.03 release, the CDP lite feature that reduces firmware resources usage is now automatically enabled when the `clockDelayPort` replication is enabled at FPGA or module level.

### Designs Using \$time are Automatically Optimized

Starting with the V-2024.03 release, the critical path timing is automatically optimized for designs using `$time` and for which `clockDelayPort` replication is enabled.

To disable automatic optimization, use the following UTF command:  
`ztopbuild -advanced_command {disable rtl_clock_ts_g_time_optimization}`

### MCP Information to Relax Feedback Paths

ZeBu compiler has limited automatic detection of MCP on feedback paths. This feature enhances the MCP detection in Fetch Mode compilations. Starting with the V-2024.03 release, this feature is enabled by default.

### MCP Information to Relax Filter-Barrier Paths

Starting with the V-2024.03 release, MCP ratio can be leveraged on filter-barrier paths (from filter-barrier flip-flop to data-domain flip-flop) by default.

### Default Transactor Fetch Mode Feature Support

Fetch Mode has been the default clock-processing mode since the S-2021.09 release. The equivalent UTF command is as follows: `performance -fetch_mode true`.

Starting with the V-2024.03 release, ZEMI3 Fetch Mode support has been set to `true` by default. The equivalent UTF command is as follows: `performance -fetch_mode true -zemi3_fm_opt true`.

For more information, refer to the **Performance** chapter of the *ZeBu® Unified Tcl Format Reference Guide*.

## Drop Asynchronous set/reset Clock Domain

The V-2024.03 release optimizes the memory usage of timing-analysis steps. Memory usage is significantly reduced on RTL designs with a large number of set/reset domains.

Use the following UTF command to configure the ZeBu timing-analysis step such that it includes set/reset domains along with the usual clock-domain information:

```
set_app_var timing_advanced_async_set_reset_analysis true
```

The default V-2024.03 behavior is equivalent to:

```
set_app_var timing_advanced_async_set_reset_analysis false
```

---

## Unified Compile Command Updates

The Unified Compile command updates are as follows:

- [Obsolete Commands](#)
- [Deprecated Commands](#)
- [New Commands](#)

## Obsolete Commands

The following commands or command options are obsolete from V-2024.03 release:

- [dpi\\_synthesis Command Options](#)

### dpi\_synthesis Command Options

Starting with V-2024.03, the following `dpi_synthesis` command options are marked obsolete:

- `-advanced_dpi <bool>`
- `-hier #filename`

## Deprecated Commands

The following commands or command options are deprecated from V-2024.03 release:

- [set\\_dualedge Command](#)
- [probe\\_signals Command Options](#)
- [blackbox Command Options](#)
- [dpi\\_synthesis Command Options](#)
- [force\\_assign Command Options](#)
- [show Command Options](#)
- [zforce Command Options](#)
- [clock\\_delay Command Option](#)
- [environment Command](#)

### set\_dualedge Command

Starting with V-2024.03, the `set_dualedge` command is deprecated.

### probe\_signals Command Options

Starting with V-2024.03, the following `probe_signals` command options are now removed and using them results in an error:

- `-clock_name <string>`
- `-exclude <string>`
- `-filter_out <string>`
- `-select <string>`
- `-size_gtr_than <int>`
- `-size_lwr_than <int>`

### blackbox Command Options

Starting with V-2024.03, the following `blackbox` command options are now removed and using them results in an error:

- `-new_module <string>`
- `-out [log|report]`

- `-select [is_terminal]`
- `-verbose`

In addition, the following values of the `<ALGO>` parameter are now deprecated:  
`disconnect`, `ignore`, `replace`.

### **dpi\_synthesis Command Options**

Starting with V-2024.03, the following `dpi_synthesis` command options are deprecated:

- `+tree #hierarchy_name`
- `-tree #hierarchy_name`
- `+path #hierarchy_name`
- `-path #hierarchy_name`

### **force\_assign Command Options**

Starting with V-2024.03, the following `force_assign` command options are deprecated:

- `-assign_as_edit <yes|no>`: Returns a deprecation warning and the option is ignored.
- `-disconnect`: Returns a deprecation warning and the option is ignored.
- `-local_driver_check`: Continues to be supported along with downstream mode
- `-source_dve <SOURCE_DVE>`: Displays an error message.
- `-only_if`: Returns a deprecation warning and the option is ignored.
- `-out <OUT>`: Returns a deprecation warning and the option is ignored.

### **show Command Options**

Starting with V-2024.03, the following `show` command options are deprecated:

- `-list`
- `<entry_name> -cycles|-type`

Use the `sniffer` and `stop` UCLI commands instead. For details about these commands, see [New Commands](#).

### **zforce Command Options**

Starting with V-2024.03, the following `zforce` command options are deprecated:

- `-pin_only`: Returns a deprecation warning and the option is ignored.

### clock\_delay Command Option

Starting with V-2024.03, the (\*ZebuClockDelay\*) parameter of the `clock_delay` command is deprecated:

It is recommended to use the corresponding UTF command `-module <module_name>`.

ZeBu displays a warning message if the attribute continues to exist in the RTL design.

### environment Command

Starting with V-2024.03, the `environment` command is deprecated. Use of this command and its options return a warning and the command is ignored:

```
environment [-generate_wrapper <NONE|C|CPP|SYSTEMC|VHDL|VERILOG>]
```

## New Commands

The following new commands are introduced in this release:

- [clock\\_boundary\\_marker](#)
- [compile Command Options](#)
- [design Command Options](#)
- [grid\\_cmd Command Options](#)
- [verilog\\_force\\_release Command Options](#)

### clock\_boundary\_marker

A new UTF command `clock_boundary_marker` is supported from V-2024.03 release.

```
clock_boundary_marker [-break] [-fnmatch] [-module <string>]  
[-object_not_found <fatal|warning>] [-rtlname <string>] [-rtlname_file  
<string>] [-signal <string>] [-verbose]
```

This command allows adding a `clock_boundary_marker` black box module in the design. This stops the clock cone propagation. Black box is removed after the clock tree annotation.

For more information about this command, refer to the *ZeBu® Unified Tcl Format Reference Guide*.

### compile Command Options

The following parameters are introduced with the `compile` command:

- `compile [-adv]`
- `compile -design_features <file>]: Specifies the name of the designFeatures file.`

- `compile [-edge_alignment_mode <int>]`: Defines `clock_delay` based support for SDC MCP computation.
- `-sdc <file>`: Specifies the name of the SDC file.
- New value for the `-objective` option: `Advance_Performance`

### **design Command Options**

The following parameters are introduced with the `design` command:

- `-enable_consistency_check <bool>`: Enables to dump all side files required for design consistency checker scalene.

### **grid\_cmd Command Options**

The following parameters are introduced with the `grid_cmd` command:

- `-cores <int>`: Specifies the number of cores used.
- `-mem_limit <string>`: Specifies the memory limit to be used.

### **verilog\_force\_release Command Options**

The following parameters are introduced with the `verilog_force_release` command:

- `-enhanced_wls <bool>`: Enables enhanced synthesis support for Verilog force statements. The options `-enable`, `-include` and `-exclude` are ignored in this mode

# 3

## New Features and Enhancements

---

This chapter discusses the new features and enhancements introduced in the following releases.

- [New Features and Enhancements in V-2024.03-1](#)
- [New Features and Enhancements in V-2024.03](#)

---

### New Features and Enhancements in V-2024.03-1

The following new features and enhancements are introduced in V-2024.03-1:

- [Compilation Stability Improvements](#)
- [Runtime Improvements](#)
- [Performance Improvements](#)
- [Debug Capabilities](#)

---

#### Compilation Stability Improvements

This section provides information about features introduced to improve compilation stability. See the following sections:

- [Consistent Naming Style for zPar Metric String IDs](#)
- [Improvements to zAudit Reporting](#)

#### Consistent Naming Style for zPar Metric String IDs

Starting with the V-2024.03-1 release, a consistent naming style is adopted for zPar metric string (attribute) IDs in the `zebu.work/tools/zMetrics/zPar_metrics.xml` file. The naming style uses uniform text separated by spaces in the IDs.

## Improvements to zAudit Reporting

Starting with the V-2024.03-1 release, the following improvements are added to zAudit reporting:

- **Summary report:** add lines for UPF, XTOR and DPI
- **New zAudit commands:**
  - `upf`: Provide detailed information about low power setting and results.
  - `dpi`: Provides list of DPI used in the design with detailed information, such as hierarchical path, number of transferred bits and number of call.
  - `xtor`: Provides the list of transactors in the design with detailed information about each XTOR, such as type, hierarchical path, number of IN and OUT ports.
  - `loop`: Provides information about SCCs, such as number of broken and unbroken loops.

---

## Runtime Improvements

This section provides information about features introduced to improve the runtime performance. See the following sections:

- [Automatically Capturing Streaming Data at Runtime](#)
- [Analog Mixed Signal Support](#)
- [Compress XML File Generation During Functional Coverage](#)
- [\\$callback Support](#)
- [C++ API for Erasing Memories](#)
- [Enable Periodic Save for Functional Coverage](#)
- [Fault Emulation](#)
- [FSDB Stitch Mode Support in zSimzilla](#)
- [Memory Word Replication Using Firmware](#)
- [Relaunching Spawned Jobs Using zSimzilla](#)
- [Reset Support in CEL](#)
- [zsvaReport Enhancement](#)



## Automatically Capturing Streaming Data at Runtime

Prior to V-2024.03-1, Runtime Trigger needs you to capture streaming data of the FSM signals. To capture streaming data of the FSM signals, you had to enable FWC or QiWC waveform capture and attach it to the `stop` command as follows:

```
set my_fid [dump -file <wave.ztdb> -fwc|-qiwc]
dump -add_value_set <value_set> -fid $my_fid
stop -cel <fsm.cel> -action <callback procedure> -fid $my_fid
dump -enable -fid $my_fid
```

Starting with V-2024.03-1, Runtime Trigger automatically captures streaming data of the FSM signals during runtime.

For details, see the *ZeBu Debug Guide*.

## Analog Mixed Signal Support

Starting with V-2024.03-1, ZeBu supports Analog Mixed Signal (AMS) and it requires hardware-specific license tokens, such as `hw_ams_zs4` or `hw_ams_zs5` to be checked out at runtime:

For details, contact Synopsys personnel.

## Compress XML File Generation During Functional Coverage

Functional coverage generates databases with size spanning 100s of MBs. This is due to the XML file present inside the generated VDB.

Starting with the V-2024.03-1 release, you can control the generation of the XML file as part of the database and compress the database when you need to keep it as a part of VDB. To do this, a new parameter, `ZEBU_UcisXmlDumpOptions`, is added to the `Zcov::SetCoverageDb()` function. Specify one of the following options for the `ZEBU_UcisXmlDumpOptions` parameter to control the XML generation:

- `ZEBU_UcisXmlDumpOptionsRemove` **removes** `ucis.xml`
- `ZEBU_UcisXmlDumpOptionsCompressed` **compresses** `ucis.xml`
- `ZEBU_UcisXmlDumpOptionsUncompressed` **saves** `ucis.xml` as it is

## \$callback Support

Starting with V-2024.03-1, `$callback` is supported with CEL to execute Runtime Trigger Tcl callback in `zRci`.

For details, see the *ZeBu Debug Guide*.

## C++ API for Erasing Memories

Starting with V-2024.03-1, a new C++ API function is introduced to erase content from given memories in batches.

```
static void EraseAll(const std::vector<Memory*> memories =  
    std::vector<Memory*>());
```

If `vector` is empty, all memories in the design are erased

## Enable Periodic Save for Functional Coverage

Starting with the V-2024.03-1 release, you can use the `Zcov::SetPeriodicDump()` function to save the Zcov database after specified cycles of `UserClock` or `TickClk`, or after a specified time interval. Counters are reset after each database is instance is saved. For parameters, see the **Performing Functional Coverage in ZeBu** chapter in the *ZeBu Functional Coverage Guide*.

To stop saving the Zcov database, use the `Zcov::StopPeriodicDump()` function. This function does not take any parameters.

## Fault Emulation

Starting from V-2024.03-1, the Fault Emulation feature is available for customers.

Fault injection mechanism is used to find potential failures that may exist within the design of a product or process. Failure mode and effect analysis can be used to identify, prioritize and limit the failures.

Fault emulation is the primary area of interest in software context that cannot be effectively covered by simulation. The solution focuses on:

- Unified fault database integration

Integration with the unified Functional Safety (FuSa) platform with interoperability of other tools for fault campaign management, including fault pruning with static/formal tools

- Performance optimization
  - Effective fault emulation cycle reduction with **zPostRunDebug** (optimization with respect to baseline emulation characteristics)
  - Real-time fault detection monitor
  - Parallel emulation in multisystem and localized replication

- The `fusa` command that was used to enable Fault Emulation is now deprecated. A new command `fault_emulation` is now provided to enable fault emulation.
- The fault's inject time can be handled using one option of the following:
  - `--inject_time`
  - Static trigger
  - Dynamic trigger

[illegible]

- A new option `--summary` aims to collect all `FaultStatus` under a specified directory into `zFaultEmuSummary.rpt` with dry run.

```
zFaultEmu --summary Fusa_offline
--fdb_import ../run_zebu/test_fusa_all.txt --campaign
fc_test_reg_scen10 --fm FM1 --testcase TC1 -z ../zcui.work/zebu.work
--start time 0 ms --end time 9 ms --sniffer frame 10
```

If there are three `FaultRun_*` directories under `Fusa_offline`, namely `FaultRun_0`, `FaultRun_1` and `FaultRun_2`, then `zFaultEmu` collects `FaultStatus` in `Fusa_offline/FaultRun 0`, `Fusa offline/FaultRun 1` and `Fusa offline/FaultRun 2`.

Starting with V-2024.03-1, you can generate ZWD and FSDB stitch-mode waveforms from the same compile database when the following UTF commands are used:

For details, see the *ZeBu Debug Guide*.

Starting with V-2024.03-1, runtime now uses a firmware feature to replicate a memory word value to an address range.

The following C++ API are impacted:

- `ZEBU::Memory::erase`
- `ZEBU::Memory::set`
- `ZEBU::Memory::clear`
- `ZEBU::Memory::loadFrom(const char* filename`

It writes into memory from the text file when the text file stores the same word value on an address range (for example @10:1000 0xcafedeca)

The following UCLI commands are impacted:

- `memory -erase <memory>`
- `memory -pattern <memory> <value>`
- `memory -fill <memory> <0|1> [-start <start_address>] [-end <end_address>]`
- `memory -fill <memory> -pattern <pattern> [-start <start_address>] [-end <end_address>]`
- `memory -load <memory> -file <filename> [-radix bin|hexa] [-start <start_address>] [-end <end_address>]`
- `memory -load <memory> -buffer <buffer> [-radix bin|hexa] [-start <start_address>] [-end <end_address>]`
- `memory -load <memory_to_file_dict> [-radix bin|hexa]`

## Relaunching Spawned Jobs Using zSimzilla

Starting with V-2024.03-1, when **zSimzilla** finished the execution of some spawned jobs that are failed, the failing jobs can be relaunched. To relaunch the failed jobs, add the `--rerun-jobs` option to the original **zSimzilla** command.

For details, see the *ZeBu Debug Guide*.

## Reset Support in CEL

Starting with V-2024.03-1, reset signal is now supported with CEL. With Runtime Trigger, the CEL reset signal now automatically resets the FSM.

## zsvaReport Enhancement

Starting with V-2024.03-1, *zsvaReport* is enhanced with the following:

- `--output sva.txt`: A new command line option is introduced and it is equivalent to online SVA API: `SetSVALog("sva.txt")`. This options prints all the error messages and a summary in the `.txt` file that is given as parameter.
- Messages printed by *zsvaReport* now include the time unit.
- Offline SVA optionally uses the “new” ZTDB format.

This feature can be enabled by using one of the following methods.

- `export ZEBU_ENABLE_NEW_ZTDB_SVA=1`
- `set_app_var ztdb_sva_api true`

### Note:

While migrating your SystemVerilog Assertion tests to V-2024.03-1, the SVA tests are impacted because time is displayed instead of ticks in the messages reported with the new ZTDB format.

For details, see the *ZeBu User Guide*.

---

## Performance Improvements

This section provides information about features introduced to improve performance. See the following sections:

- [Support for Variable Delay in RTL Clock Replication](#)
- [Reshaping of User Memories](#)

### Support for Variable Delay in RTL Clock Replication

In the earlier releases, the *clock delay* feature in ZeBu did not support replication of variable delays in every FPGA (or core). The replication feature was limited to constant delays or `clockDelayPorts`. As a result, RTL clock delay replication could not be used if delays were defined as variable registers (to reduce timing paths) in the RTL.

Starting with the V-2024.03-1 release, you can replicate variable delay clocks in an RTL. The UTF command to enable RTL clock replication is as follows:

```
clock_delay -module <clk_generator_module_name>
            -enable_var_delay_replication true
```

For more information about the UTF command, see the *ZeBu UTF Reference Guide*.

## Reshaping of User Memories

Starting with the V-2024.03-1 release, the RTL memories can be reshaped in the backend. This saves capacity, could also ease partitioning, and avoid issues such as congestion and overflow further down the line.

Use the following **zTopBuild** UTF command to enable the reshaping user memories:

```
ztopbuild -advanced_command {memory_opt port_opt -enable_resaping}
```

For more information about reshaping user memories, see the Memory Optimization section in the **Compilation** chapter of the *ZeBu User Guide*.

---

## Debug Capabilities

This section has the following topics:

- [Support for VHDL Modules in Toggle Coverage](#)

## Support for VHDL Modules in Toggle Coverage

Starting with the V-2024.03-1 release, Toggle Coverage supports VHDL modules in ZeBu flow.

---

## New Features and Enhancements in V-2024.03

The following new features and enhancements are introduced in V-2024.03:

- [Compilation Stability Improvements](#)
- [Runtime Improvements](#)
- [Debug Capabilities](#)
- [New C++ Classes and APIs](#)

---

## Compilation Stability Improvements

This section provides information about features introduced to improve compilation stability. See the following sections:

- [Error Displayed When Unified XMR Flow is Disabled](#)

## Error Displayed When Unified XMR Flow is Disabled

Unified XMR flow is enabled by default. VCS displays an error if the switch to disable unified XMR flow is detected in the VCS command.

The following error message is displayed:

```
Error-[NATIVE_XMR_FLOW_NOTSET] Native XMR Flow is not enabled. Please  
enable Native XMR Flow as non-Native flow is not supported.
```

---

## Runtime Improvements

This section provides information about features introduced to improve the runtime performance. See the following sections:

- [Identifiers Added for Runtime Warnings](#)
- [Checkpoint and Restart Features](#)
- [Sample Control in zRci](#)
- [zTune: Printing Hardware Information](#)
- [zTune: New Markers for Enhanced Performance](#)
- [zdpiReport Enhancement](#)
- [zRci Options](#)

## Identifiers Added for Runtime Warnings

Starting with V-2024.03, warnings at runtime now have identifiers like warning during compilation. For example, the following warning has changed.

Example:

Runtime Warning before V-2024.03:

```
zServer : WARNING : The controlled clock  
"U0.M0.hw_top.clockGen15.clock"'s waveform is set to default "_-".
```

Runtime Warning with V-2024.03:

```
zServer : WARNING : LUI3919W : The controlled clock  
"U0.M0.hw_top.clockGen15.clock"'s waveform is set to default "_-".
```

## Checkpoint and Restart Features

Starting with V-2024.03, the following enhancements are added to Checkpoint and Restart features:

### 7-zip Utility to be included in ZeBu Package

The 7-zip utility is delivered with Zebu and is available in the installation directory at `$ZEBU_ROOT/thirdparty/dmtcp/7Zip`.

The 7-zip utility provides the most optimum compression and decompression speed and ratio when used with DMTCP for checkpointing the software state of emulation.

## Sample Control in zRci

Starting with V-2024.03, the `wap` command is introduced to enable and set sample control based on the policy provided.

```
wap -dump -enable [-policy  
<average|performance|balance|accuracy|precision>] [-sampling_ratio 1|2]
```

For details, see the *ZeBu Unified Command-Line User Guide*.

## zTune: Printing Hardware Information

Starting with V-2024.03, the `report_hardware` command in zTune is enhanced to print hardware metadata and hardware component details. The new options are as follows:

- Hardware Metadata: `report_hardware --meta`
- Hardware Component: `report_hardware --id ID`

For details on these options, see *ZeBu Runtime Performance With zTune Application Notes*.

## zTune: New Markers for Enhanced Performance

Starting with V-2024.03, on the left-hand side of the zTune GUI, there is a new **Markers** sub-panel . This panel displays features with potential impact on performance. To enable a feature, in the right-hand side Markers sub-panel, under the **System hierarchy**, select the corresponding check box.

## zdpiReport Enhancement

Starting with V-2024.03, the `-delete-processed-fwc` parameter is introduced in the `zdpiReport` tool to delete the `raw_fwc_*` files from ZTDB when the `zdpiReport` tool is running in on-fly-mode.

For details, see the *ZeBu User Guide*.

## zRci Options

The following zRci options are introduced in this release:

- `sniffer`
- `stop`



### sniffer

Sniffer technology enables you to record the ZeBu state and save the stimuli. Specify the `sniffer` UCLI command to use the Sniffer technology. The syntax for the `sniffer` command is as follows:

```
sniffer -info [<name>] [-cycles|-time]
```

For more information about this command and its options, see the *ZeBu UCLI User Guide*.

### stop

Use the new `stop` UCLI command to manage runtime triggers. The syntax for this command is as follows:

```
stop -info -events [<entry>] -cycles|-time{code}
```

For more information about this command and its options, see the *ZeBu UCLI User Guide*.

---

## Debug Capabilities

This section has the following topics:

- [Vega2 is Enabled for the Verdi Flow](#)
- [zSimzilla Enhanced to Improve Faster Waveform Conversion](#)

### Vega2 is Enabled for the Verdi Flow

Verdi flow has been improved to use common elaboration with VCS. Starting with the V-2024.03 release, this flow is supported in ZeBu with the `ZCUI_EXPERT` environment variable. To enable this feature, set `ZCUI_EXPERT` to `enableVega2ForVerdi` before invoking ZeBu.

If pre-compiled libraries are used, then these should be regenerated by adding "`-kdb=common_elab`" to the analysis commands.

### zSimzilla Enhanced to Improve Faster Waveform Conversion

Starting with V-2024.03, the `zSimzilla --capture-only` command runs parallel jobs based on the following criteria: first, based on the number of FPGAs and second is based on ZTDB slicing.

Earlier, you had to execute the `--co-by-fpga 2` option with the `zSimzilla --capture-only` command to enable running parallel jobs based on the criteria.

## New C++ Classes and APIs

Starting with V-2024.03, the following C++ classes and APIs are added:

Class	APIs Added
Board	<ul style="list-style-type: none"> <li>• void Board::selectSignalsToRandomize(const char *signalList, const ZEBU_Randomizer_FileType&amp; type);</li> <li>• void Board::excludeSignalsToRandomize(const char *signalList = 0, const ZEBU_Randomizer_FileType&amp; type = ZEBU_Randomizer_FileType::TypeSignalList);</li> <li>• void Board::excludeObjectsToRandomize(const std::vector&lt;Filter*&gt;&amp; filters);</li> <li>• void Board::selectObjectsToRandomize(const std::vector&lt;Filter*&gt;&amp; filters);</li> </ul>
CCall	<ul style="list-style-type: none"> <li>• static void CCall::ReadOfflineDpiSpecification(Board *board, const std::list&lt;const char*&gt; filenames);</li> <li>• static void CCall::ClearOfflineDpiSpecification(Board *board);</li> </ul>
CompatibilityChecker.hh	
Waveform	<ul style="list-style-type: none"> <li>• Waveform::WaveformCaptureConfig <b>class</b></li> <li>• Waveform::ProtoConfig <b>class</b></li> <li>• void setConfig(const Waveform::ProtoConfig&amp; config)</li> <li>• void setConfig(const WaveformCaptureConfig&amp; config);</li> </ul>
FastWaveformCapture	static int SetIdleMaskMode(int mode);
SVA	static void EnableMessageReporting(Board *board, bool enable);
PowerProfile	static void clearInstance();
XtorBase	<ul style="list-style-type: none"> <li>• void resetLowActivityImports()</li> <li>• void terminateLowActivityImports()</li> </ul>
ZEMI3Handler	<ul style="list-style-type: none"> <li>• bool getIsLowActivityImport(void) const;</li> <li>• void setIsLowActivityImport(void);</li> </ul>

Class	APIs Added
ZEMI3Manager	<ul style="list-style-type: none"><li>• void terminateLowActivityImports();</li><li>• void resetLowActivityImports();</li></ul>
ZEMI3Xtor	<ul style="list-style-type: none"><li>• void resetLowActivityImports</li><li>• void terminateLowActivityImports</li></ul>

# 4

## LCA Features

---

This release introduces the following LCA features. For LCA features pertaining to previous releases, see the *ZeBu LCA Features Guide*.

- [LCA Features in V-2024.03-1](#)
- [LCA Features in V-2024.03](#)

---

### LCA Features in V-2024.03-1

The LCA features introduced in V-2024.03-1 are explained in the following sections:

- [ECO Flow for Dynamic Triggers](#)
- [ZRM Performance Mode is Enabled by Default](#)

---

#### ECO Flow for Dynamic Triggers

Starting with V-2024.03-1, when using Dynamic Triggers in your design, ECO Flow is supported to modify an existing Dynamic Triggers or create a new Dynamic Trigger using the ECO transactor architecture. When you specify new input to Dynamic Triggers, the inputs are connected ECO transactors and Dynamic Triggers are calculated at workstation.

For details, see the *ZeBu Limited Customer Availability (LCA) Features Guide*.

---

#### ZRM Performance Mode is Enabled by Default

Starting with V-2023.03-1, the latency optimization of ZRM memories (`zrm_performance_mode`) is enabled by default. When enabled, the `zrm_performance_mode` reduces the latency of the ZRM memories by 140ns. The enhanced runtime impacts runtime throughput by 8%.

For more information, see the *ZeBu LCA Features Guide*.

---

## LCA Features in V-2024.03

The LCA features introduced in V-2024.03 are explained in the following sections:

- [Checkpoint and Restart Using DMTCP](#)
- [Toggle Coverage](#)

---

### Checkpoint and Restart Using DMTCP

The checkpointing feature allows you to save the status of the entire hardware DUT and the software testbench environment during the emulation runtime. The most common scenario is to avoid using valuable emulation cycles to get the previous system start/initialization when running software tests.

When hardware state is natively captured by ZeBu Save and Restore functionality, saving the software testbench environment is done using an external open-source tool called Distributed MultiThreaded Checkpointing (DMTCP).

---

### Toggle Coverage

Toggle Coverage is an effective method for ensuring wide-spread coverage of the testbench and helps in achieving verification closure. This feature enables capturing Toggle Coverage information in the VDB format. You can use Verdi or Unified Report Generator (URG) in VCS to view and analyze coverage reports.

Following features are supported with Toggle Coverage:

- Various VCS controls including `cm_hier` to enable or disable coverage in various modules/hierarchies
- Support for MDAs
- C++ control to enable/disable coverage collection at runtime
- Support for Simulation Acceleration flow
- Merging of coverage databases using Verdi or URG

For more information about the Toggle Coverage feature, see the *ZeBu LCA Features Guide*.

# 5

## Deprecated Features

---

This section describes deprecated features in the following releases:

- [Deprecated Features in V-2024.03-1](#)
- [Deprecated Features in V-2024.03](#)

---

### Deprecated Features in V-2024.03-1

Deprecated features are explained in the following sections:

- [Deprecated Databases](#)

---

#### Deprecated Databases

Starting with V-2024.03-1, RTLDB and ZRDB are marked as deprecated databases. ZeBu only supports MuDb and only RTL names can be used with ZeBu.

---

### Deprecated Features in V-2024.03

Deprecated features are explained in the following sections:

- [Hardware Updates](#)
- [C++ APIs and Classes](#)

---

#### Hardware Updates

Starting with S-2021.09-2, ZeBu Server 3 is deprecated and the following error message is displayed for `zSetupSystem/zUtils` and runtime.

```
zUtils :  
- Zebu : zUtils : ERROR : ZUT1594E : The zUtils utility no longer  
  supports the Zebu Server 3 platform. Contact Synopsys support to get a  
  release still supporting this platform.
```

zServer:  
RTPUB0005E : ZeBu-Server 3 platform is not supported anymore. Contact  
Synopsys support to get a release supporting this hardware.

---

## C++ APIs and Classes

Starting with V-2024.03, the following C++ classes and APIs are deprecated:

Classes/ API Deprecated	Comments
<code>Driver::getSignal()</code>	Use <code>Board::getSignal()</code> instead.
In the Waveform class, <code>UIP</code> type	Use <code>SequentialWaveformCapture</code> type
In the Waveform class, <code>Sequential</code> type	Use <code>SequentialWaveformCapture</code> type
In the Board class, <code>-Board::selectSignalsToRandomize(const char* signalList, const ZEBU_Randomizer_FileType&amp; type)</code> <code>onst char *signalList = 0)</code>	Use <code>Board::selectSignalsToRandomize(const char* signalList, const ZEBU_Randomizer_FileType&amp; type)</code>

# 6

## Obsolete Features

This section describes obsolete features in the following releases:

- [Obsolete Features in V-2024.03](#)

### Obsolete Features in V-2024.03

The following features are obsolete from V-2024.03:

- [Obsolete APIs](#)

#### Obsolete APIs

Starting with V-2024.03, the following C++ APIs are marked as obsolete:

*Table 1      Obsolete APIs*

API/Class	Comments
<code>ZEBU::PartSignal</code>	<p>This API has been removed as “deprecated” since S-2021.09.</p> <p>This class contained two public methods:</p> <ul style="list-style-type: none"> <li>• <code>* ZEBU::PartSignal::Create()</code></li> <li>• <code>* ZEBU::PartSignal::TryToCreate()</code></li> </ul> <p>Starting with V-2024.03, the following error is displayed if you try to use it:</p> <pre>-- Zebu : tb : ERROR : ZPUB8641E : PartSignal API isn't supported anymore.</pre>
<code>Board::SignalIterator</code>	
<code>Driver::SignalIterator</code>	
<code>FreqMeter</code>	<code>FreqMeter.hh</code> header is also removed.
<code>ZCTraceMemory</code>	<code>ZCTraceMemory.hh</code> header is also removed.
<code>ZEMI3</code>	<p>In <code>ZEMI3</code> class, extern "C" void* <code>ZEMI4_getExpHndlBySvScpAndId(void* scope, char* TFName);</code></p>



Table 1      *Obsolete APIs (Continued)*

API/Class	Comments
Memory	In <b>Memory class</b> , void <code>setMemoryAccessMode (uint32_t width);</code>
Sniffer	In <b>Sniffer class</b> , static bool <code>CheckUnsupported(const char *featureName);</code>
zTune	<p>In <b>zTune class</b>, the following:</p> <ul style="list-style-type: none"><li>• -static void <code>SetOutputPath(const char* ztuneDumpPath = 0, bool dumpCSV = true)</code><ul style="list-style-type: none"><li>◦ -Use <code>Init(Configuration)</code> and <code>Start(const char*)</code></li></ul></li><li>• -static void <code>Start(Configuration config)</code><ul style="list-style-type: none"><li>◦ Use <code>Init(Configuration)</code> and <code>Start(const char*)</code></li></ul></li></ul>

# 7

## Limitations

---

This chapter lists limitations for various ZeBu components in the following topics:

- [Operating Systems](#)
- [Compilation Interface \(zCui\)](#)
- [Compilation](#)
- [Runtime](#)
- [SystemVerilog Assertions \(SVAs\)](#)
- [zPostRunDebug](#)
- [ZEMI-3](#)

---

### Operating Systems

There are no known limitations related to the OS versions used for the current release.

---

### Compilation Interface (zCui)

For more information, see the following subsections:

- [zCui uses /bin/sh and ignores .cshrc Aliases or Functions](#)
- [Limitations in the Compilation View](#)

---

#### **zCui uses /bin/sh and ignores .cshrc Aliases or Functions**

**zCui** uses `/bin/sh` and ignores aliases and functions defined elsewhere (for example, in `.cshrc`). Ensure that your scripts use the executable names instead of the aliases.

---

#### **Limitations in the Compilation View**

At the end of each compilation, the final FPGA parameters are stored in a file located in the Back-End directory. However, this file is rewritten after each compilation, leading to the

loss of the previous FPGA parameters. These parameters cannot be retrieved and reused for subsequent compilations.

---

## Compilation

For more information, see the following subsections:

- [Unsupported Performance Oriented Partitioning with Direct ICE](#)
- [Unsupported Timing Optimization Features](#)
- [Memory Modeling](#)

---

### Unsupported Performance Oriented Partitioning with Direct ICE

When the Direct ICE interface is used in the verification environment, it is not possible to use **Performance Oriented Partitioning** (**Clustering** panel). It results in errors during zCore-level compilation.

---

### Unsupported Timing Optimization Features

The following timing optimization features are not supported by the partitioner along with its **zTopPartitioning** and **zCorePartitioning** tools:

- **Post Partitioning and Partitioning Driven** options in the **Clustering** panel > **Automatic Generation of zCoresframe** (**System** workspace) in **zCui**.
- The following commands of the **zTopClustering** and **zCoreClustering** tools:
  - `-set_async_optim_strength`
  - `-set_cutobjective`
  - `-set_global_false_path`
  - `-enable_async_full_optim` (equivalent to **Partitioning Driven** option in **zCui**)
  - `-enable_async_optim` (equivalent to **Post Partitioning** option in **zCui**)

---

## Memory Modeling

### Limitation on Read/Write Priority for Multi-port zrm Memories

The read-before-write priority is set individually on a port. The read/write priority between two ports of the same memory instance is not deterministic for `zrm` memories, even if both ports use the same clock signal.

## Automatic Memory Loading

If you want to load your memory at runtime with `readmemb/readmemh`, you should not use `implementAsFlops`; instantiate a `zMem` memory instead.

When the `implementAsFlops` attribute is used, the initialization of the memory must be done during synthesis with memory content in-lining.

---

## Runtime

This section describes the emulation runtime limitations. For more information, see the following subsections:

- [Initialization Order in a C++/C Testbench](#)
- [Limitation on Fast Hardware State](#)
- [Disk Resource Conflicts for ZTDB Files](#)
- [Memory Initialization not Carried out in Some C/C++ Testbenches](#)
- [Clock Declaration in the designFeatures File](#)
- [Message Size in Clock-Delay Mode](#)

---

### Initialization Order in a C++/C Testbench

In a C++/C testbench, for cosimulation or transaction-based emulation, it is mandatory to initialize the message ports after calling the `Board::open` method (or `ZEBU_Board_open` function) but before calling with the `Board::init` method (or `ZEBU_Board_init` function).

Note that the methods/functions to connect the message ports differ with the verification environment:

- `Driver::connect` method (or `ZEBU_Driver_connect` function) for C++/C cosimulation.
- `Port::connect` method (or `ZEBU_Port_connect` function) for transaction-based environments.

If the order of initialization is not correct, the emulation stops with the following error message (the identifier of the error might be `ZHW0810E` or `ZHW0761E` according to your verification environment):

```
Unable to connect the port <driverName>::<portName>.
You must register the port just between the open and the init functions.
```

---

## Limitation on Fast Hardware State

A fast hardware state object cannot be initialized before board initialization.

See the following topics for more details:

- [Sampling Clocks for Waveform Capture](#)

## Sampling Clocks for Waveform Capture

Previous versions of the ZeBu software allowed the selection of sampling clocks used for waveform capture. However, this selection of the sampling clock silently interfered with the proper behavior of the following features:

- `zInject`, used by `zPostRunDebug`
- `zForce` for release synchronization of forced values
- Direct ICE for input synchronization
- UPF Features
- DPI calls
- SVAs

This release ignores requests to change the waveform capture sampling clocks, and issues a warning message in the testbench log file.

Sampling should always be at the dual-edge of the composite clock.

### Impact

You see the following impact with this new behavior:

- The waveforms that use non-default sampling clocks may appear different as compared to the waveforms from the previous releases. Note that both waveforms are accurate.
- The size of the waveform files might increase.
- The runtime performance might decrease when capturing waveforms.

### Impacted APIs and their C/Tcl Equivalents

The impacted API functions and their C and Tcl equivalents are:

- `FastWaveformCapture::SelectSamplingClocks`
- `CCall::SelectSamplingClockGroup`

- `Call:SelectSamplingClocks`
- `SVA::Start`

---

## Disk Resource Conflicts for ZTDB Files

When capturing waveforms, and using Post-Run Debugging ZTDB format, several files with this extension coexist and the ZeBu runtime library might create several hard links for them.

If these files are located on different disks/partitions (local partition or remote resources), the following error might be displayed:

```
ERROR : ZPRIV0062E : Cannot create link "[<second path>/]<second
name>.ztdb/<space file>" on "[<first path>/]<first name>.ztdb/<space
file>", Invalid cross-device link
```

To avoid this error, it is recommended to declare file system paths on the same disk/partition for all waveform output files and for the sniffer directory.

---

## Memory Initialization not Carried out in Some C/C++ Testbenches

When a C/C++ testbench controls the design without using transactors or cosimulation drivers (`HDL_COSIM`, `C_COSIM`, `MCKC_COSIM`), some memories are not initialized. In this case, the design does not work even if no error is generated.

To avoid this, users must identify the testbench in the `designFeatures` file in the same way as is done for multi-process.

```
$nbProcess = 1;
$process_0 = "#process_name";
```

Note that the `#process_name` field must match the 3rd parameter of the open call (`zebu.work`, `designFeatures`, `process_name`). This parameter is not mandatory; the default can be used:

```
$nbProcess = 1;
$process_0 = "default_process";
```

At the beginning of the connection, an unidentified process generates the following warning messages:

```
-- ZeBu : cpp_test_bench : WARNING : A process name has been specified
"default_process" at open time, but it is not specified in the
"./designFeatures" file.
-- ZeBu : cpp_test_bench : WARNING : The list of specified process names
is :
-- ZeBu : cpp_test_bench : Looking for a connection (pid 28241 at Tue 3 8
2010 - 17:46:57)
```

```
-- ZeBu : cpp_test_bench : "default_process" is a control-only process  
working on "../zebu.work".
```

An identified process issues the following:

```
-- ZeBu : cpp_test_bench : Looking for a connection (pid 28200 at Tue 3 8  
2010 - 17:46:11)  
-- ZeBu : cpp_test_bench : "default_process" is a full-capability process  
working on "../zebu.work".
```

Unidentified processes can be used to investigate or control clocks, memories, logic analyzers, signals, and so on. They cannot be used to control top-level I/Os.

---

## Clock Declaration in the designFeatures File

### Specification of the Initial Clock State

Unexpected initialization results on registers or memories, or mismatch with simulation results occur when a controlled clock is specified to start at high level.

To inform users that a non-recommended initial state is declared, the following warning message is displayed in the **zServer** log:

```
-- ZeBu : zServer : WARNING : The controlled clock "U0.M0.AAAA"'s  
waveform is "-_", it's not recommended to start clocks with a high  
level.
```

Clocks starting at a low level are strongly recommended to avoid the effects of the invisible rising edge when starting emulation.

---

## Message Size in Clock-Delay Mode

Messages between the testbench process and **zServer** have a fixed size (1024 bytes).

In some cases, this size is too small, and the following error message is displayed:

```
-- ZeBu : zServer : ERROR : LUI2706E : Client/server message too long  
followed by the message size and the message content (truncated)
```

In particular, this can happen in clock-delay mode when the user describes complex waveforms in the `designFeatures`.

---

## SystemVerilog Assertions (SVAs)

This section describes SVA limitations. For more information, see the following subsections:

- [Unsupported SVA Features](#)

---

### Unsupported SVA Features

#### Full Report Mode

Full Report Mode is not supported.

#### Report only Failure Mode

Specific limitations for **Report only Failure** mode are the following:

- **Sequences:**  
Task/function calls with local variables as arguments are not supported.
- **Concurrent Assertions:**
  - Multi-clocked assertions are not supported in procedural blocks.
  - Assertions with a clocking event different from the clocking event of the enclosing procedural block are not supported

---

## zPostRunDebug

This section describes **zPostRunDebug** limitations. For more information, see the following subsections:

- [Limitation on Primary Clocks](#)
- [Limitation with Direct-ICE and SMART Z-ICE Clocks](#)

---

### Limitation on Primary Clocks

16 primary design clocks can be stored by the sniffer, provided they are declared in a single group.



---

## Limitation with Direct-ICE and SMART Z-ICE Clocks

When the emulation environment uses the SMART Z-ICE or Direct-ICE interfaces, it is not possible to use **zPostRunDebug** because of the input clocks coming from these interfaces.

An environment using the Direct ICE interface can be used with **zPostRunDebug** only when output clocks are controlled by the ZeBu clock generator.

---

## ZEMI-3

The model name of a transactor cannot be named with a name conflicting with the ZEMI-3 public API method member.

## 8

## Fixed STAR List

---

The following table lists the STARs fixed in this release:

4808734	4818118	4881520	4767035	5084185	5094792	5087725	5102787	5102331	5110243
5108323	5074256	5077135	4814016	4784329	4797675	5494438	5493020	5491139	5421263
5420211	5431959	5431921	5430960	5421023	5419845	5419863	5423744	5423435	5423438
5416540	5432003	5427246	5447807	5446567	5443892	5443831	5443710	5433945	5432172
5432006	5431956	5067292	5057888	5045692	5039230	5051280	5046843	5029857	5029883
5069838	5095581	5124475	5027482	5118756	5116383	5115844	5122474	5108682	5107710
5109305	5108334	5117716	5112545	5111404	5111406	5105134	5062055	5056190	5058651
5058561	5056281	5071089	5070697	5068279	5067056	5019490	5006880	5014960	5006753
5037175	5000557	4986338	4993317	5042247	5068098	5044991	5095222	5092090	5091845
5090420	4970938	4849586	4849507	4849089	4849542	4840306	4880492	4871680	4943631
4962515	4967526	4896968	4900836	4862208	4858371	4847463	4849088	5401120	5397367
5393448	5406008	5404422	5402440	5414503	5414593	5414196	5413961	5403805	5410407
5414079	5398444	5320996	5319742	5333154	5319036	5325371	5319799	5347071	5345270
5333142	5316582	5269764	5269735	5256201	5271550	5261595	5257490	5258245	5258114
5258074	5255792	5262669	5262063	5262017	5262093	5257097	5258862	5303857	5302411
5300067	5308482	5308720	5280232	5278067	5276750	5273672	5294358	5291087	5293654
5279762	5276141	5276138	5275301	5277954	5277354	5277147	5279627	5279919	5356448
5350937	5350285	5310326	5309000	5308490	5307993	5303267	5303409	5306873	5307366
5308511	5333224	5333222	5328854	5324456	5324449	5328952	5342995	5335793	5332801
5333147	5288731	5283205	5286615	5289248	5288931	5284062	5283550	5283534	5282460
5282334	5283789	5283422	5283207	5283158	5283166	5275404	5274477	5266055	5266059

5264046 5269176 5297224 5292627 5296932 5305592 5302878 5295177 5294796 5319917  
5318141 5337909 5335313 5335782 5335723 4679993 4670316 4670327 4661820 4626659  
4643245 5535849 5531108 5569892 5537300 5535454 5530407 5526786 5521274 5524456  
5527219 5521191 5527947 5524578 5524502 5546315 5544258 5543691 5552744 5548779  
5559645 5544982 5542302 5541872 5538936 5553138 5549330 5502310 5502050 5501307  
5499471 5499489 5509701 5498638 5506791 5506500 5505636 5505330 5503978 5516548  
5506062 5505858 5505262 5514906 5499836 5503976 5503602 5521061 5512728 5520173  
5515205 5514382 5514176 5514115 5507668 5504875 5503493 5503496 5520102 5516521  
5516587 5494435 5491159 5488091 5489007 5488848 5555554 5556937 5556819 5558964  
5529277 5529731 5540970 5543644 5532895 5531110 5529977 5531790 5532157 5529995  
5529837 5510883 5495404 5499246 5518360 5514017 5510706 5509798 5506863 5511535  
5496824 5494253 5491116 5489204 5494626 5493997 5493050 5492831 5492526 5492701  
5489435 5489464 5489654 5489208 5489384 5488183 5502331 5548412 5547945 5544262  
5541410 5544915 5407008 5399155 5397004 5396268 5394894 5392122 5391696 5391575  
5391173 5381913 5377489 5375405 5377091 5375543 5376099 5433943 5447225 5428940  
5426968 5429534 5426653 5450205 5449897 5449833 5449802 5448328 5443926 5434913  
5431947 5431894 5426879 5426888 5449308 5447485 5447459 5447176 5447244 5446495  
5442774 5371072 5369931 5372954 5366378 5365727 5373359 5371307 5373987 5366389  
5372316 5372346 5360877 5363286 5362227 5361309 5358023 5358152 5364195 5362219  
5361030 5363630 5362010 5359606 5365170 5384733 5383967 5377871 5377454 5393186  
5387953 5387444 5387482 5384941 5388722 5357376 5350933 5349984 5356497 5355816  
5353596 5351934 5354813 5362465 5356792 5354875 5354258 5353665 5356601 5349625  
5349546 5349968 5424804 5419856 5414520 5414515 5412167 5406591 5408084 5407957  
5400548 5407273 5406802 5341730 5341720 5353750 5351907 5349615 5341522 5342968  
5415280 5451276 5451807 5424917 5424820 5426848 5414301 5145435 5241471 5237707  
5242024 5140646 5137666 5132381 5136209 5134055 5131077 5121300 5120978 5127982  
5123067 5133692 5133294 5120945 5113514 5244401 5241656 5237784 5254924 5251972

5240950 5238481 5238215 5255063 5148279 5147680 5146711 5141920 5148708 5234868  
5234714 5096682 5100402 5109967 5096859 5112522 5098280 5124786 5137262 5137185  
5145767 5139185 5138244 4994176 4991802 4992599 5002668 4917232 5015093 5015721  
5020244 5013219 5045440 5022035 5046507 5082351 5078068 5244746 5444549 5434379  
5449649 5449513 5389938 5389989 5397977 5392367 5392110 5516577 5515555 5514024  
5514018 5240773 5237601 5279922 5278064 5276684 5275760 5274121 5247296 4470394  
4593397 4587772 4116431 4232992 4232996 4608072 4587361 4647160 5419556 5567695  
5134081 5394182 5115539 5557513 5585574 5590160 5568526 5371866 5360735 5559085  
5514028 5497928 5393661 5328801 5580556 5516532 5391905 5545662 5414356 5578287  
5514816 5555562 5581170 5581158 5320841 5496480 5510815 5510801 5510298 5561729  
5530173 5442002 5573429 5601690 5576805 5308777 5290813 5119472 5544458 5311455  
5566537 5574703 5527204 5533275 5591739 5559760 5588185 5561924 5042358 5042343  
5441942 5524744 5569216 5555438 5364729 5386023 5298685 5507956 5405373 5548631  
5499147 5515215 5286722 5559753 5559490 5568129 5042939 5574263 5563408 5581976  
5544266 5559805 5585061 5504049 5546036 5514022 5552864 5126117 5505303 5550371  
5542309 5541869 5525950 5542305 5493435 5511540 5412346 5283306 5551445 5551443  
5527244 5526386 5563464 5449637 5449603 5533176 4243001 4949069 5450891 5516536  
4522108 5559957 5600612 5548482 5098279 5503649 5139192 5523452 5520652 5531806  
5450224 5526565 5528058 5526528 5543283 5556957 5544722 5564430 5412385 5422765  
5511573 5547979 5549061 5571440 5425210 5097138 5613468 5529838 5536151 5406362  
5499500 5049319 5577960 5590808 5351127 5530367 5393181 5514019 5547011 5562023  
5551365 5434006 5495628 5519833 5364179 5527051 5271748 5081671 5571461 5103551  
5404647 5571959 5320157 5384337 5539811 5442059 5393192 5413349 5352682 5283377  
5577305 5566956 5385386 5560122 5341118 5125058 5547981 4227821 5569045 5512822  
5560120 5560121 5560123 5558180 5503975 5447027 5134530 5504208 5011757 5533290  
5568793 5394033 5568471 4968736 5514378 5394031 5536343 4874372 5376375 5552328  
5544200 5505595 5411047 5556636 5557650 5243806 5446585 5573331 5392245 5558516

5533947 5505616 5497466 5425443 5317129 5522594 5489670 5320936 5513266 5493687  
5494903 5330670 5497628 5343034 5361478 5379944 5393004 5515562 5313605 5325541  
5344095 5410401 4585735 4436712 5567251 5581451 5563616 5361507 4978134 5347169  
5260904 4860927 5612029 5505788 5567451 5534072 5529281 5601734 5256358 5598171  
5567645 5525287 5617595 5559593 5556134 5260483 5498868 5608577 5514178 5370444  
5612419 5612412 5420558 5390372 5315523 5528324 5496138 5603357 5571672 5627848  
5011809 5096629 5533339 5434057 5491180 5629547 5591442 5605648 5589683 5421511  
5590426 5132399 5346529 5536497 5599831 5598446 5637110 5066163 5068641 5147673  
5251434 5571913 5410547 5407280 5598980 5012822 5609260 5514318 5339013 5287382  
5591423 5527630 5578083 5488363 4968115 5533189 4833352 5273678 5569530 5620421  
5433972 5529820 5603560 5520376 5614915 5596789 5416572 5500107 5401096 5557665  
5557649 5431515 5561875 5535895 5551631 5480937 5490459 4945292 5424406 5425502  
5503396 5608328 5416431 5642726 5614960 5514786 5614342 5548467 5137231 5594650  
5328426 5425130 5584580 5596562 5557285 5514025 5522388 5582866 5629941 5556750  
5617594 5622723 5575391 5304596 5591927 5536502 5367907 5503995

---

# 9

## Known Issues

---

This chapter describes version compatibility, and consists of the following subsections:

# 10

## Version Compatibility

---

This chapter describes version compatibility, and consists of the following subsections:

- [Operating System Compatibility](#)
- [Hardware Compatibility](#)
- [Checking the Need for a New Xilinx License File](#)
- [Xilinx Vivado License for ZeBu Server 4 Hardware](#)
- [New Version of xilind License Daemon for ZeBu Server 4 Hardware](#)
- [Synopsys Interoperable Technologies](#)
- [Third-Party Tools Compatibility](#)

---

### Operating System Compatibility

See the [Operating Systems](#) section for details about operating system compatibility.

---

### Hardware Compatibility

This release is compatible with ZeBu Server 4.

---

### Checking the Need for a New Xilinx License File

License files from Xilinx contain a version limit for each feature/increment in the license. The Xilinx Version Limit corresponds to a year and a month (for example, 2010.04 corresponds to April 2010), as described on the Xilinx website.

---

### Xilinx Vivado License for ZeBu Server 4 Hardware

The ZeBu Server V-2024.03-1 software ships with Vivado 2022.1, which requires a 2022.04 Version Limit or higher.

If the Xilinx Version Limit of your license file is not 2022.04 or higher, you must request a new Vivado license file from Synopsys.

To determine whether you need a new Vivado license file or not, perform the following steps:

1. Open the Xilinx license file, typically `XILINX.lic`.
2. Look for the line containing `INCREMENT Logic_Edition` with the version limit.

In the following example, the Version Limit is 2022.04:

```
INCREMENT Logic_Edition xilinxd 2022.04 permanent 1 EE39502AC1CB \
```

This license can be used with Vivado 2022.01, which is delivered with ZeBu Server V-2024.03-1.

In the second example, the Version Limit is 2013.12:

```
INCREMENT Logic_Edition xilinxd 2013.12 permanent 1 956BC4E5D360 \
```

Here the license cannot be used with Vivado 2022.1, which is delivered with ZeBu Server version V-2024.03-1.

---

## New Version of xilinxd License Daemon for ZeBu Server 4 Hardware

Since Vivado 2021.1 (which first shipped with ZeBu 2021.09), Vivado requires a new version of the `xilinxd` license daemon.

If Vivado cannot check out its license due to an older version of the `xilinxd` license daemon, the following warning message is displayed in the Vivado log file:

```
WARNING: [Common 17-348] Failed to get the license for feature  
'Implementation' and/or device 'xcvu440'.
```

For details, set the environment variable as `FLEXLM_DIAGNOSTICS=3` and rerun Vivado.

An additional message is displayed in the server log file, as follows:

```
2:38:46 (xilinxd) Request denied: Client (11.14) newer than Vendor  
Daemon (11.13). (Version of vendor daemon is too old. (-83,21049))
```

You must replace the existing `xilinxd` daemon with the new `xilinxd` daemon and restart the license server. The new `xilinxd` daemon is located in `$ZEBU_ROOT/vivado_p2/bin/unwrapped/linux64.o/xilinxd` and restart the license server.



## Synopsys Interoperable Technologies

The following table lists the Synopsys interoperable technologies tested with the present version of the ZeBu software.

Table 2      *Synopsys Interoperable Technologies*

Tool	Tested Versions	Remarks
VCS MX	V-2023.12-SP1-1-TZ	HDL Simulator
VCS Native Low Power (NLP)	V-2023.12-SP1-1-TZ	Synopsys Low Power Verification Tools Suite
SpyGlass	V-2023.12-SP1-1-TZ	
Verdi	V-2023.12-SP1-1	Debug tool for waveform capture and expansion
ZeBu Empower	V-2023.12-SP4	
Synopsys DesignWare Building Block	S-2021.06-SP4	
Transactors and Memory Models	V-2024.03	
Virtual System Adaptors	V-2024.03	
Hybrid Adaptor Library	V-2024.03	
ICE Speed Adaptors	S-2023.09-1	

\* OS compatibility might change. For more information, see <http://www.synopsys.com>.

**Note:**

The Synopsys interoperable tools for ZeBu can be downloaded from SolvNetPlus.

## Third-Party Tools Compatibility

The following table lists third-party tools that are tested with the present version of the ZeBu software.

Table 3 Third-Party Tools

Tool	Tested Versions	Tested OS Compatibility *	Remarks
SystemC	2.2.0	RHEL 7.3+ RHEL 8+	Required for SystemC cosimulation: <a href="http://www.systemc.org">http://www.systemc.org</a>
Gcc	9.5	RHEL 7.3+ RHEL 8+	C compiler. Part of the Linux operating system distribution. Required for C/C++ and SystemC co simulation and for compilation of the software part of the transactor and of the runtime testbench. <a href="http://gcc.gnu.org">http://gcc.gnu.org</a> Includes binutils 2.33.1 and GDB v8.3
ZeBu Server 4:ZeBu_Vivado _2022.1_patch d18	RHEL 7.3+ RHEL 8+	<a href="http://www.xilinx.com">http://www.xilinx.com</a>	Included in the ZeBu software package.

\* OS compatibility might change.

## GCC Version and Recommendations

GCC Version and command line options:

- The `-pthread` option on the `GCC/g++` command line is mandatory when you compile a testbench or a dynamic library.

The ZeBu V-2024.03-1 release has been compiled using `GCC/g++ 9.5` with the `--std=c++17` option. Therefore, it is recommended to:

- Use this `GCC/g++` version or higher to compile `C++` testbenches and dynamic libraries. Otherwise, the following warning message is displayed:

```
-- ZeBu : zRci: WARNING : ZRCI0988W : Testbench was compiled using an
unsupported compiler. Consider using GNU g++ 9.5.0
```

- Use the `--std=c++17` option on the `g++` command line.

# 11

## Documentation

---

ZeBu documentation is delivered in HTML and PDF formats, which you can access from SolvNetPlus or the ZeBu installation area.

For more information, see the following topics:

- [Accessing ZeBu Server 4 Documentation in SolvNetPlus](#)
- [Documentation in Installation Area](#)
- [Additional Navigation in ZeBu Documentation](#)

---

### Accessing ZeBu Server 4 Documentation in SolvNetPlus

You can access ZeBu Server 4 documentation by either entering keywords in the SolvNetPlus search bar or by navigating to the ZeBu documentation page. See the following topics:

- [Logging Into SolvNetPlus](#)
- [Using Search](#)
- [Viewing the ZeBu Server 4 Documentation Page](#)
- [Downloading the ZeBu Server 4 Documentation Package](#)

---

### Logging Into SolvNetPlus

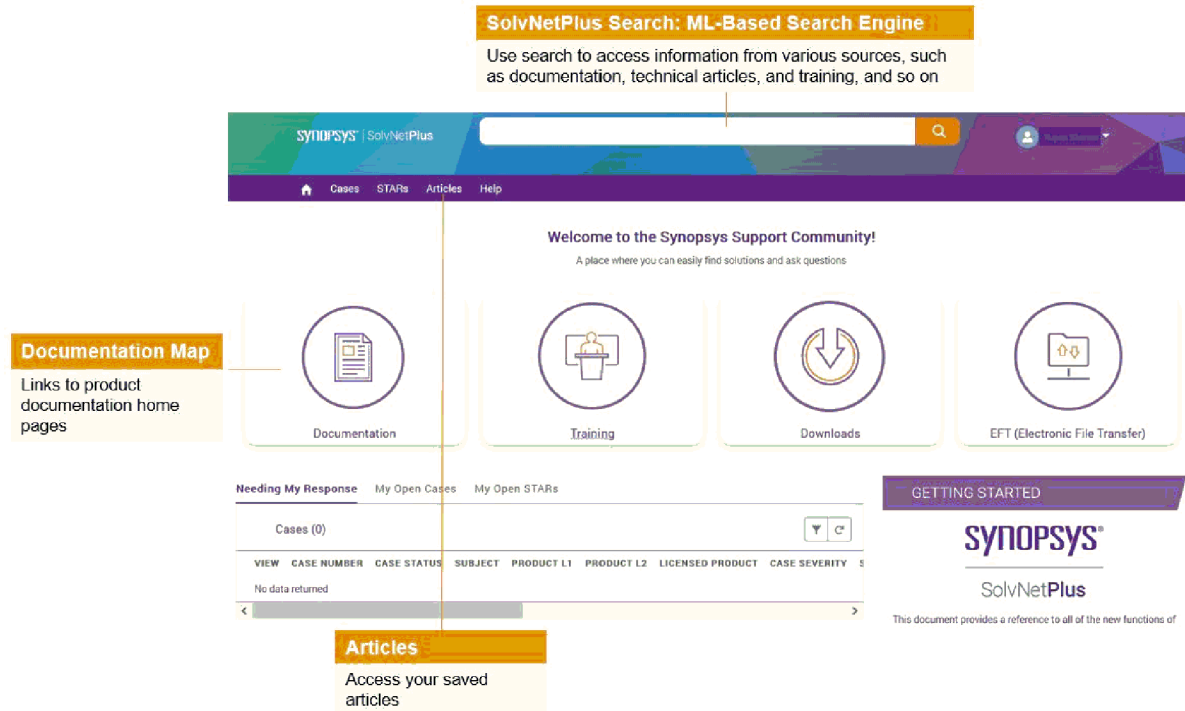
To access the documentation, log into SolvNetPlus using your SolvNetPlus login credentials. The SolvNetPlus URL is as follows:

<https://solvnetplus.synopsys.com/s/>

**Note:**

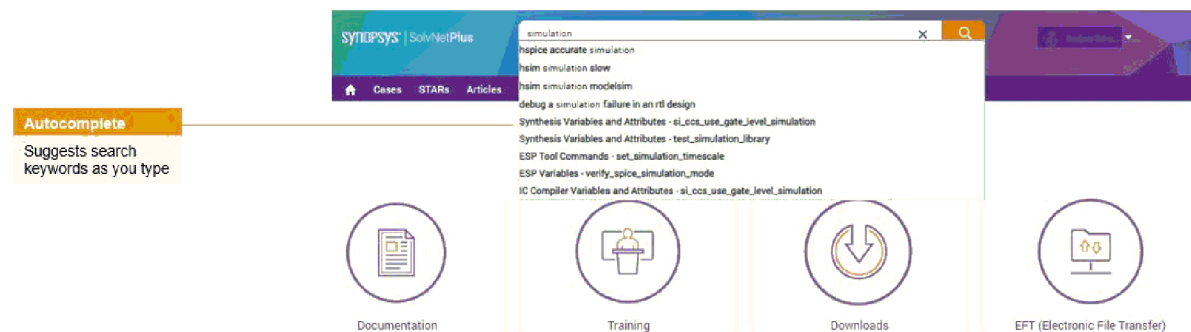
It is recommended to use the Chrome, Firefox, and Safari browsers for accessing SolvNetPlus.

SolvNetPlus provides an advanced search engine that you can use to retrieve information from various sources, such as documentation, articles, cases, STARs, training, YouTube, and so on. The following figure shows the SolvNetPlus Home page.



## Using Search

To look for information, enter search keywords in the search bar on the home page. As you enter the search keywords, relevant search strings get displayed, as illustrated in the following screenshot.



The SolvNetPlus Search page displays the search results. You can refine these results based on your requirement using the various filters, as shown in the following figure.

The screenshot shows the SolvNetPlus search results page. On the left, there are three main sections: **Source Tabs**, **Dynamic Filters**, and **Source Labels**. **Source Tabs** includes checkboxes for Cases (26,755), Docs - Silicon Tools (118,927), Articles (8,791), Docs - Silicon IP (3,794), Cases (1,995), and YouTube (22). **Dynamic Filters** includes checkboxes for VCS (75,209), VeriQuest R (77,350), VeriQuest Compiler (44,872), SayData (98,438), Unisim (16,713), Saker (13,112), Design Compiler (10,014), and Jovial Compiler (8,773). **Source Labels** includes checkboxes for Search (2,115), Virtualizer (741), and Release (34,681). The main search results area shows a list of documents with columns for Title, Product, and Release. Annotations point to various features: **Source Tabs** (Choose the sources that you would like to search in), **Dynamic Filters** (Populates search filter options dynamically based on search context), **Source Labels** (Clearly identify the source in the search results), **Quick View** (Preview the content without clicking the search result), and **Email** (Share the content link with other users).

You can refine search results by using the **Source Tabs** and **Dynamic Filters**.

For more information, log into SolvNetPlus and see the [SolvNetPlus FAQ page](#) under Help.

## Viewing the ZeBu Server 4 Documentation Page

Apart from using the search bar, you can access ZeBu Server 4 documentation by navigating to the ZeBu documentation page. The following figure shows how to access the ZeBu Documentation landing page.

The diagram illustrates the navigation path from the SolvNetPlus Home Page to the ZeBu Documentation Product Landing Page. It consists of three main steps: **Click Documentation on SolvNetPlus Home Page**, **Click ZeBu in the product list**, and **ZeBu Documentation Product Landing Page**. The first step shows the SolvNetPlus Home Page with a 'Documentation' link highlighted. The second step shows the 'Documentation' page with a 'ZeBu' link highlighted. The third step shows the 'ZeBu Documentation Product Landing Page'.

On the ZeBu Documentation landing page, you can access the following:

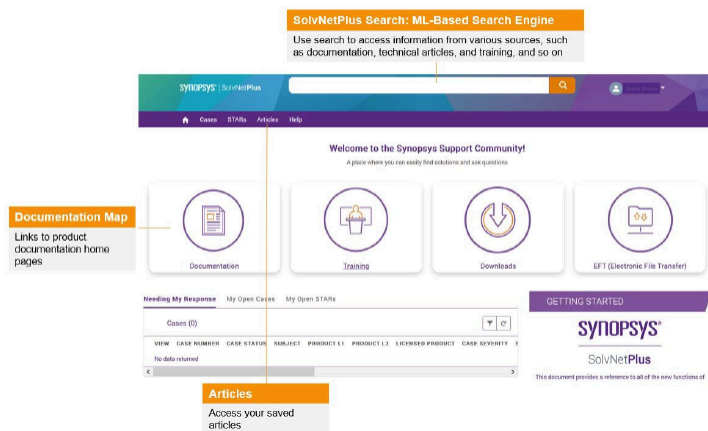
- **Getting Started:** Provides a block diagram, highlights, and additional resources. Hover over the blocks in the block diagram to view infotip. For more information on a block, click the block.
- **PDFs:** Provides access to PDFs pertaining to ZeBu Server 4. For easier legibility, the PDFs are categorized.
- **FAQs:** Provides access to commonly asked questions. Click the question to see the answer in the relevant page of the ZeBu Server 4 HTML help set.

Click the **Feedback** button to provide feedback on documentation to the Technical Publications team.

## Downloading the ZeBu Server 4 Documentation Package

After the **Synopsys Installer** software is downloaded and extracted, download the ZeBu Documentation Package from SolvNetPlus.

1. Log on to the Synopsys SolvNetPlus website (<https://solvnetplus.synopsys.com>) with your credentials.
2. Click the **Downloads** link on the page.



3. From the **Downloads** list, choose **ZeBu Server All > V-2024.03-1**.
4. On the next page, click **Download Here**. The **Electronic Software Transfer terms and conditions** are displayed.
5. Click **AGREE and SIGN IN** at the end of **Electronic Software Transfer** terms.

The **Synopsys Electronic Software Transfer (EST)** page opens.

6. Download the following `.spf` package for installing ZeBu Server 4 documentation:

```
zebu_docs_V-2024.03-1_common.spf
```

7. After you have downloaded the package, move it to the directory containing the Synopsys Installer software.
8. To install the ZeBu Server 4 documentation using a script, open a terminal and execute the following commands:

```
$ cd /home/<user>/install_files  
$ ./installer
```

The script mode of the Synopsys Installer is launched. You must fill in all the requested parameters for the installation. For more details, see “Installing the ZeBu Software”, in the *ZeBu Site Administration Guide*.

---

## Documentation in Installation Area

See the following subsections:

- [Accessing ZeBu Documentation Using SmartSearch](#)
- [Accessing ZeBu Documentation in HTML Format](#)
- [Accessing ZeBu Documentation in PDF](#)

---

## Accessing ZeBu Documentation Using SmartSearch

Apart from accessing ZeBu Server documentation in PDF and HTML formats, you can now access the documentation using the ZeBu SmartSearch tool. This tool is an advanced search engine for retrieving information from the ZeBu Server documentation. The search engine technology leverages natural language processing (NLP) and machine learning techniques to provide you with relevant search results.

ZeBu SmartSearch provides the following capabilities:

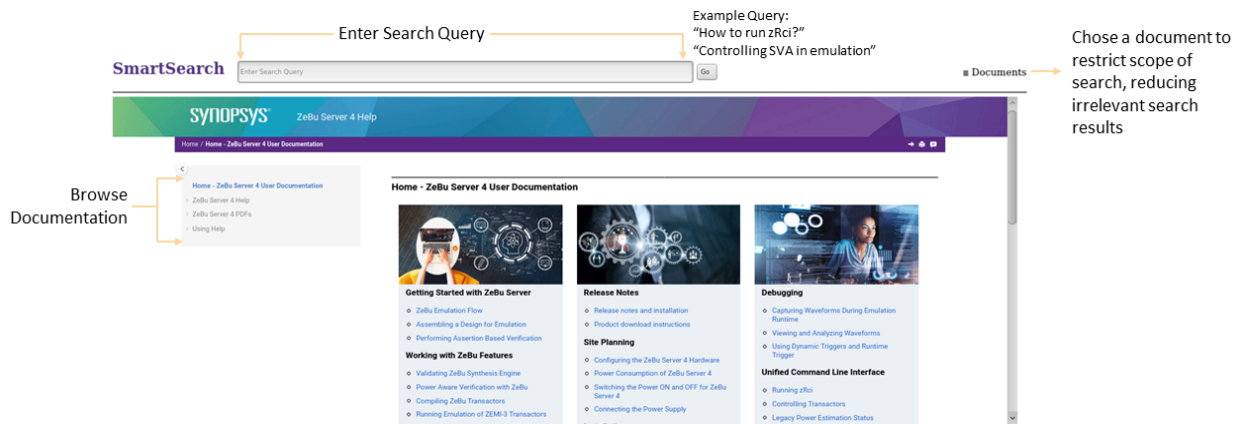
- Advanced information search techniques that enables you to find relevant information faster
- Handles search queries in natural language
- Learns from user interaction and maintains history of searches

To launch SmartSearch, run the following command:

```
zCui --doc
```

The following figure shows the key elements of the SmartSearch interface.

Figure 1 SmartSearch Tool



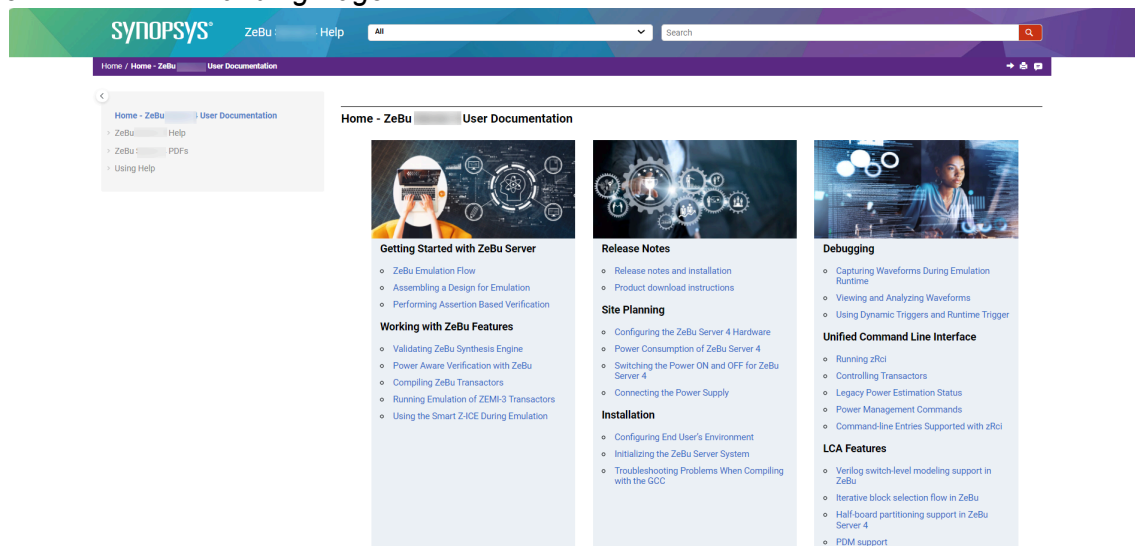
The Technical Publications team appreciates your feedback on SmartSearch. To channel feedback, send a mail to [support\\_center@synopsys.com](mailto:support_center@synopsys.com) with the Subject **"Feedback on [ZeBu][Documentation][ZeBu Server SmartSearch]"**.

## Accessing ZeBu Documentation in HTML Format

The `$ZEBU_ROOT/doc/HTML` directory contains all ZeBu Server documentation in HTML file format. You can view these files using a web browser in Linux or Windows.

To view the HTML files, open the `index.html` file located in the `$ZEBU_ROOT/doc/HTML` directory.

Figure 2 HTML Landing Page

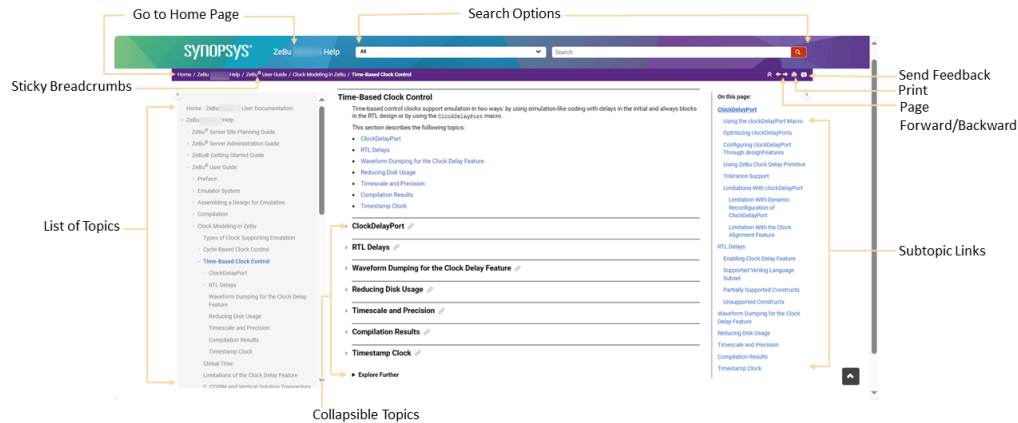




## Key Features

The following figure displays the HTML Help interface and the available options.

Figure 3 HTML Help Interface



The key features of HTML help are follows:

### New Look and Feel

The new look and feel of the HTML help is designed to improve user experience and be consistent with the Synopsys SolvNetPlus website design.

### Responsive Tile Layout

The new responsive tile layout auto-fits the page content and appearance based on the screen size and orientation of the device you are using.

### Cleaner Group Categories

On the Table of Contents (TOC), the HTML and PDF files are reorganized with cleaner group categories, making it easier to scan for information you want to read.

### Sticky Breadcrumbs

Breadcrumb navigation shows you where you are in the help's hierarchy.

Site navigation is now always on the top of the page as you scroll up and down – improving the findability of WebHelp sections and pages.

This helps you to find your way around, especially when hiding TOC (to increase the content area in the browser window).

### Collapsible Topics

Collapsible topics allow organizing the content of a long page in a simpler structure, giving the benefits of less scrolling and quick scanning for the topic of interest.

To reduce page scrolling and quickly scan for the topic of interest,

- Use the **Collapse sections** button to compress page contents with multiple topics in a simpler structure. Then, expand the topic you want to read.
- On the other hand, if you want to expand all topics again, click the **Expand sections** button.

### Subtopic Links

For a long page with multiple subtopics, the help also offers subtopic links on the right of the page so that you can quickly zoom in on the topic of interest and go exploring.

### Return to Landing Page

Quick ways to return to the landing page:

- Click **Home** on the sticky breadcrumb
- Click the product name displayed on the title bar

### Search Across Documentation Suite

Search for information across the documentation by selecting *All* in the drop-down list and entering the search keywords in the **Search** box. The search is case-insensitive.

The search returns the relevant instances from the complete ZeBu documentation collection.

You can also limit your search to a specific document by clicking on the drop-down list and selecting a specific documentation.

To channel feedback to the Technical Publications team, click the **Email** icon.

---

## Accessing ZeBu Documentation in PDF

The `$ZEBU_ROOT/doc/PDF/` directory contains all ZeBu Server documentation in PDF file format. You can view these files using a PDF reader in Linux or Windows.

### Viewing PDF Files

To view PDF files, perform the following steps:

1. Open the `navigator.pdf` file located in the `$ZEBU_ROOT/doc/PDF/` directory. The `navigator.pdf` file contains links to the PDF documents located in the `$ZEBU_ROOT/doc/PDF/` directory.
2. Click the document link to open the PDF.

## Searching in All PDFs

If you are searching for specific information, but are not sure which PDF file contains the relevant information, you can use the Search feature. The Search feature enables you to search across all documents in the `$ZEBU_ROOT/doc/PDF/` directory.

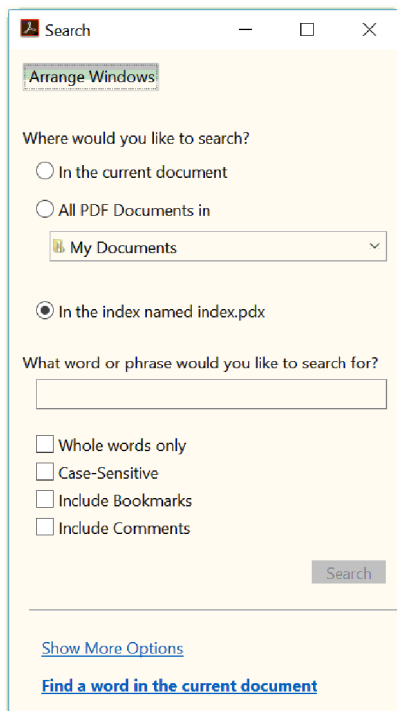
To search all PDFs, perform the following steps:

1. Open the `Index.pdx` file. The **Search** window appears.

### Note:

If the Search window does not appear on a Linux machine, associate the `pdx` file extension with Acrobat.

The following example figure displays the **Search** window.



2. Enter the search keywords in the **What word or phrase would you like to search for?** box.
3. By default, the search is performed in the index named `Index.pdx`. This index file has been created based on the all PDF files located in the `$ZEBU_ROOT/doc/PDF/` directory.
4. By default, a basic search is performed. Select the check boxes to refine the search.

By default, the search returns all instances in the complete ZeBu documentation collection that match the exact word or phrase entered in Step 2.

5. To perform an advanced search, specify additional search criteria under **Show More Options**.
6. Click **Search** to see each instance of the search keywords entered in Step 2. The instances are grouped by the name of the PDF.

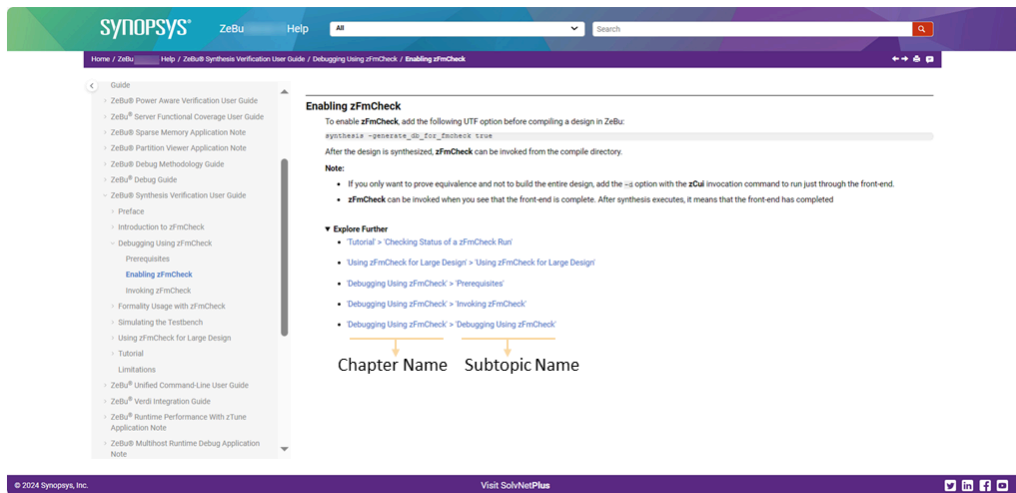
To channel feedback to the Technical Publications team, click the **Feedback** link located in the footer of the PDF pages.

## Additional Navigation in ZeBu Documentation

Links to relevant topics have been added to most pages in the HTML format. These links are located under the *Explore Further* heading, as shown in [Figure 4](#).

The *Explore Further* links provide one-click access to topics sourced from within a guide and across the entire help set. For example, the following screen shot shows the *Debug Commands* page in the *ZeBu Server UCLI Guide*, the *Explore Further* links all point to topics, which might be useful, in the *ZeBu Server Debug Guide* or the *ZeBu Server Debug Methodology Guide*.

Figure 4 *Explore Further Links*



To know more about how to see ZeBu Server documentation in HTML format, see [Documentation in Installation Area](#).