Migrating from ZeBu-XXL to ZeBu-Server

AN032
Revision a

ZeBu Application Note

July 2010

Purpose:

This document presents useful recommendations to migrate a working design and its environment from ZeBu-XXL to Zebu-Server.

This document deals with installation, compilation and emulation runtime.

For compilation in particular, this document explains how to migrate a V4_3_3 **`zCui`** compilation project to V6_2_1B and explains how to migrate from a script-based V4_3_3 compilation to **`zCui`** in V6_2_1B.

Applicability:

This document is applicable for ZeBu software version 4_3_3D (ZeBu-XXL existing environment) and version 6_2_1B (ZeBu-Server targeted software).

History:

This table gives information about the content of each revision of this manual, with indication of specific applicable version:

Doc Revision	Date	Evolution
a	Jul 2010	First Edition

Copyright Notice
Proprietary Information

Right to Copy Documentation

The license agreement with EVE permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any. Licensee must assign sequential numbers to all copies. These copies shall contain the following legend on the cover page:

"*This document is duplicated with the permission of EVE, for the exclusive use of* ____________________________ *and its employees. This is copy number* ______."

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

EVE 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.

Table of Contents

Figures

Tables

1 Functional Compatibility

1.1 Compilation Flow

The compilation flow has been largely modified between V4_3_3 and V6_2_1B, in particular for back-end compilation: the mapping step (zBuild in V4_3_3) is now a 2-step process for system-level compilation (**zTopBuild**) in which one or several zCores declared for the design in order to process medium-size netlists in the zCore-level compiler (**zCoreBuild**). Because of this new compilation flow, it is no longer possible to compile with user-written scripts in V6_2_1B.

This document presents how to migrate from a V4_3_3 **zCui** compilation project to a V6_2_1B compilation project (in Section 2.1) and how to migrate a **zBuild** script for V4_3_3 compilation to a V6_2_1B **zCui** compilation project (in Section 2.3).

When synthesizing with **zFAST**, the V6_2_1B compilation flow supports RTL paths declaration in the DVE file and for instance- and signal-related commands, as described in *ZeBu-Server Release Note for V6_2_1B*.

1.2 Direct ICE

The Direct ICE interface of ZeBu-Server is mostly compatible with ZeBu-XXL one, with a reduced number of global clocks (only 4 global clocks available in ZeBu-Server).

However, the commands which were processed by **zBuild** in V4_3_3 are in a separate script declared in **zCui** in V6_2_1B. Their syntax has been modified.

You should refer to the *ZeBu-Server Direct ICE Manual* for further details about the supported feature and the syntax of the commands. A separate document should be available soon to provide dedicated support to migrate a ZeBu-XXL Direct ICE environment to ZeBu-Server.

1.3 EVE Vertical Solution Packages

When the verification environment includes EVE Vertical Solutions products, transactors or memory IPs, the following points should be checked to migrate from ZeBu-XXL to ZeBu-Server:

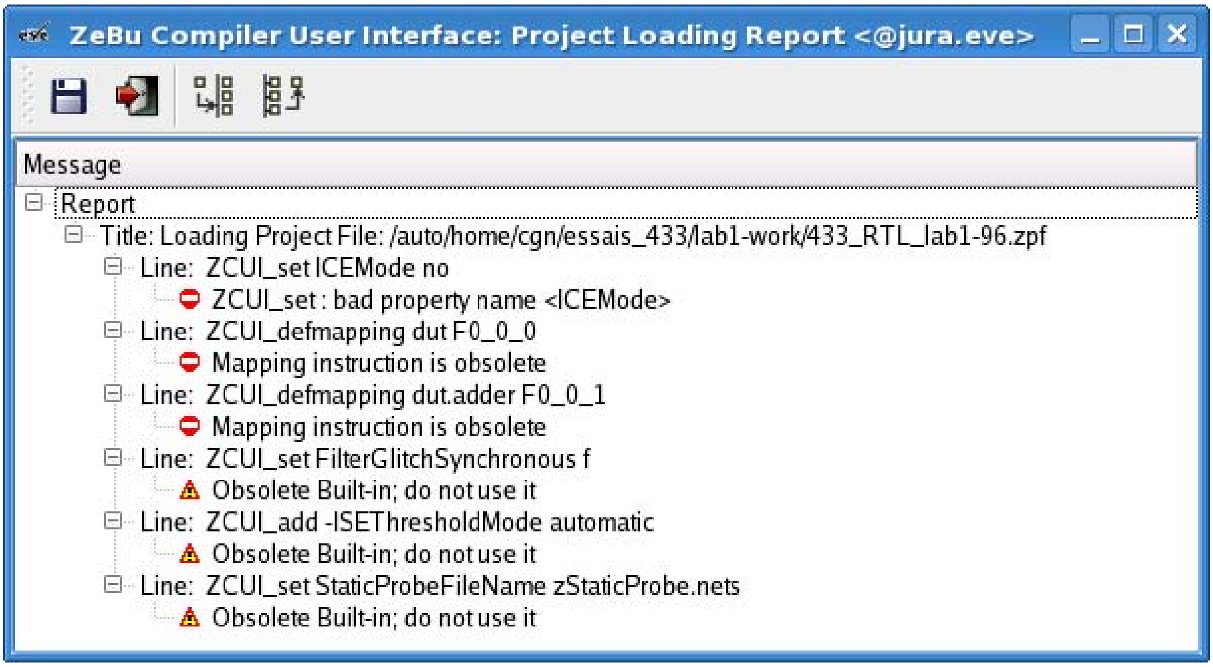
- Memory IPs: All memory IPs are compatible between ZeBu-XXL and ZeBu-Server but the NAND Flash IP requires a specific package for ZeBu-Server.
- Some transactors have specific packages for ZeBu-Server, with the same features as equivalent 64-bit packages:
  - GMII Ethernet Transactor: use `GMII_ZSE_64bit.2.0.tgz` package
  - I2C Transactor: use `I2C_ZSE_64bit.2.0.tgz` package
  - PCI Express Transactor: use `PCIE_ZSE_64bit.5.0.tgz` package
  - USB Transactor: use `USB_ZSE_64bit.1.2.tgz` package

2 Modified User Inputs

2.1 zCui Project

2.1.1 Messages when opening the compilation project

When opening a compilation project, `zCui` detects the erroneous settings and shows them in a separate window:



ZeBu Compiler User Interface: Project Loading Report <@jura.eve>

```
Message
⊟ Report
    ⊟ Title: Loading Project File: /auto/home/cgn/essais_433/lab1-work/433_RTL_lab1-96.zpf
        ⊟ Line:  ZCUI_set ICEMode no
              ⛔ ZCUI_set : bad property name <ICEMode>
        ⊟ Line:  ZCUI_defmapping dut F0_0_0
              ⛔ Mapping instruction is obsolete
        ⊟ Line:  ZCUI_defmapping dut.adder F0_0_1
              ⛔ Mapping instruction is obsolete
        ⊟ Line:  ZCUI_set FilterGlitchSynchronous f
              ⚠ Obsolete Built-in; do not use it
        ⊟ Line:  ZCUI_add -ISEThresholdMode automatic
              ⚠ Obsolete Built-in; do not use it
        ⊟ Line:  ZCUI_set StaticProbeFileName zStaticProbe.nets
              ⚠ Obsolete Built-in; do not use it
```

Figure 1: `zCui` messages when importing V4_3_3 project

The reported messages show the content of the lines of the zpf file which cannot be interpreted by `zCui`:

- **Obsolete Built-in; do not use it**:
  The corresponding item in the `zCui` interface has been removed or its name has changed. In most cases, the name of the built-in is very similar to the name displayed in the graphical interface. User should refer to the corresponding snapshot in the following sections to see how to configure the compilation in V6_2_1B.

- **Mapping instruction is obsolete**:
  When user-defined clustering was selected in V4_3_3, the corresponding `defmapping` commands are no longer valid in V6_2_1B (because the system is different and because the clustering process has changed). See Section 2.1.3.5, Table 3 and Table 4 for detailed information.

When the default value of a `zCui` option has changed between V4_3_3 and V6_2_1B, it is imported with its V4_3_3 value when loading the V4_3_3 compilation project in V6_2_1B.

2.1.2 Sources

2.1.2.1 Declaration of probes

With V4_3_3 software, the declaration of probes was done in a dedicated Tcl script which was declared in the **Sources → Design → Probe Files** item of **zCui**. It was automatically taken into account when synthesizing the design and also processed by **zBuild**.

With V6_2_1B software, the declaration of probes is also done in a Tcl script but the name of the **zCui** item has changed: **Sources → Design → RTL-based Compilation Scripts**. When synthesizing with **zFAST**, this script may include other compilation commands, in addition to the probe declaration: `force`, `force_dyn` and `tristate` (see Section 2.3).

When importing a V4_3_3 existing compilation project, the scripts declared in the **Probe Files** item are automatically added in the **RTL-based Compilation Scripts** item.

2.1.2.2 RTL-based Compilation Scripts

Please refer to the ***ZeBu-Server Release Note - V6_2_1B*** for a complete description of this feature.

2.1.2.3 Synthesizing with **zFAST**

Please refer to the ***ZeBu zFAST Synthesizer Manual*** and to the ***ZeBu-Server Release Note - V6_2_1B*** for details.

2.1.2.4 Synthesizing with a third-party synthesizer

When a third-party synthesizer is selected, the **Main** panel is the same in V6_2_1B as in V4_3_3.

In the **Synthesizer** tab, the default value for the **Synthesis Mode** has changed but it is imported with its V4_3_3 value when loading the V4_3_3 compilation project in V6_2_1B.

The other tabs are unchanged between V4_3_3 and V6_2_1B.

2.1.2.5 Memory Sources

2.1.2.6 EDIF Sources

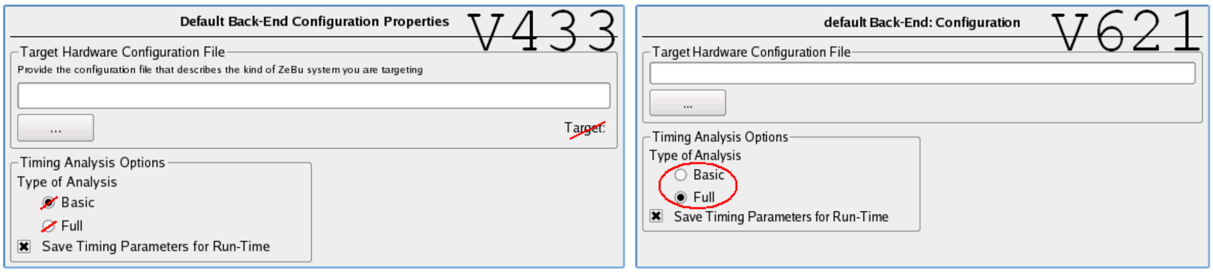
With ZeBu-Server, it is possible to use EDIF sources whioch result of Virtex-4 synthesis for ZeBu-XXL. However, the improvement in the FPGA technology will not be available when the design was synthesized for Virtex 4.

This capability may be useful to avoid synthesizing a project which was synthesized from RTL sources

2.1.3 Back-End Compilation

2.1.3.1 Configuration

The items in the **Back-end: Configuration** panel are the same in V4_3_3 and V6_2_1B.



Default Back-End Configuration Properties
V433

Target Hardware Configuration File
Provide the configuration file that describes the kind of ZeBu system you are targeting

~~Target:~~

Timing Analysis Options
Type of Analysis
● Basic
○ Full
☒ Save Timing Parameters for Run-Time

default Back-End: Configuration
V621

Target Hardware Configuration File

Timing Analysis Options
Type of Analysis
○ Basic
● Full
☒ Save Timing Parameters for Run-Time

Figure 2: ***Back-End*** **→** ***Configuration*** **panel**

The **Target Hardware Configuration File** which was declared in V4_3_3 should not be kept with V6_2_1B since it now describes a Zebu-Server system.

If a V4_3_3 configuration file is declared when compiling with V6_2_1B, it causes an error in the **Build System** step:

```
# Configuration used is:
#   valid machines   : default,
#   created by       : EVE
#   date             : Sat Jul  3 16:41:13 2010
#   based on release : V4_3_3D
#   platform type    : UF
### Error: Cannot access file
/auto/common/zebu/V6_2_1.100615/etc/sys/ZUF/configs/zuf_basic_conf.tcl. Please check your
installation.
### tcl error in zTopBuild [ZTB0005F] : couldn't read file
"/auto/common/zebu/V6_2_1.100615/etc/sys/ZUF/configs/zuf_basic_conf.tcl": no such file or
directory
```

The default setting for **Timing Analysis Options → Type of Analysis** has changed to **Full**. The compilation project is imported with the V4_3_3 value when loading the V4_3_3 compilation project in V6_2_1B.

2.1.3.2 Clock Declaration

No modification in this **zCui** panel. The possible values for the **Type** column are the same in both versions.

2.1.3.3 Environment

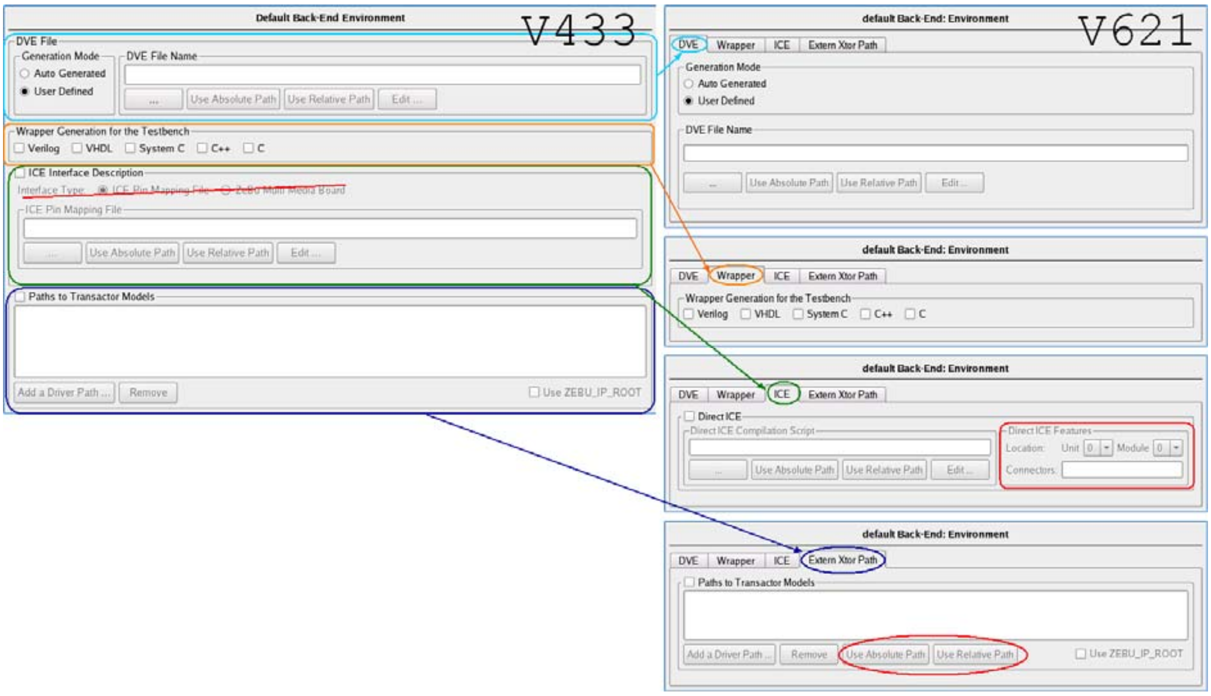

Default Back-End Environment
V433
DVE File
Generation Mode
○ Auto Generated
● User Defined
DVE File Name
... Use Absolute Path Use Relative Path Edit ...
Wrapper Generation for the Testbench
□ Verilog □ VHDL □ System C □ C++ □ C
□ ICE Interface Description
Interface Type ● ICE Pin Mapping File ○ ZeBu Multi Media Board
ICE Pin Mapping File
Paths to Transactor Models
Add a Driver Path ... Remove
□ Use ZEBU_IP_ROOT
default Back-End: Environment
V621
DVE Wrapper ICE Extern Xtor Path
Generation Mode
○ Auto Generated
● User Defined
DVE File Name
Wrapper Generation for the Testbench
□ Verilog □ VHDL □ System C □ C++ □ C
□ Direct ICE
Direct ICE Compilation Script
Direct ICE Features
Location: Unit 0 Module 0
Connectors:
□ Paths to Transactor Models
Add a Driver Path ... Remove Use Absolute Path Use Relative Path
□ Use ZEBU_IP_ROOT

Figure 3: ***Back-End → Environment* panel**

This panel has been modified for better legibility: the frames that were displayed in V4_3_3 have been replaced by 4 tabs with roughly the same information.

The most significant modifications are visible in the **ICE** tab, because the Direct ICE feature is supported differently between Zebu-XXL and Zebu-Server, as described in a separate Application Note. Details on how to fill the Direct ICE frame are available in the Zebu-Server Direct ICE Manual.

When importing a V4_3_3 existing compilation project, all other settings of the **Environment** tab are correctly set in this new interface.

2.1.3.4 Core Definition

This tab has been introduced for ZeBu-Server compilation flow in order to declare 2 compilation scripts:

- **Core Definition File**: declaration of the zCores (`defcore` command), target module declarations for relocation (`use_module` command), FPGA allocation (`use_fpga` command).
- **Netlist Edition File**: declaration of design blackbox (`blackbox` command), individual register initialization (`reg_init` command).
  In case some commands which should be in the **Netlist Edition File** are included by mistake in the **Additional Command File**, they will be executed

correctly and a warning message will be issued. Ideally, user should move these commands to the appropriate file for future compilations.

2.1.3.5 Clustering

The **Clustering** panel has been largely modified:

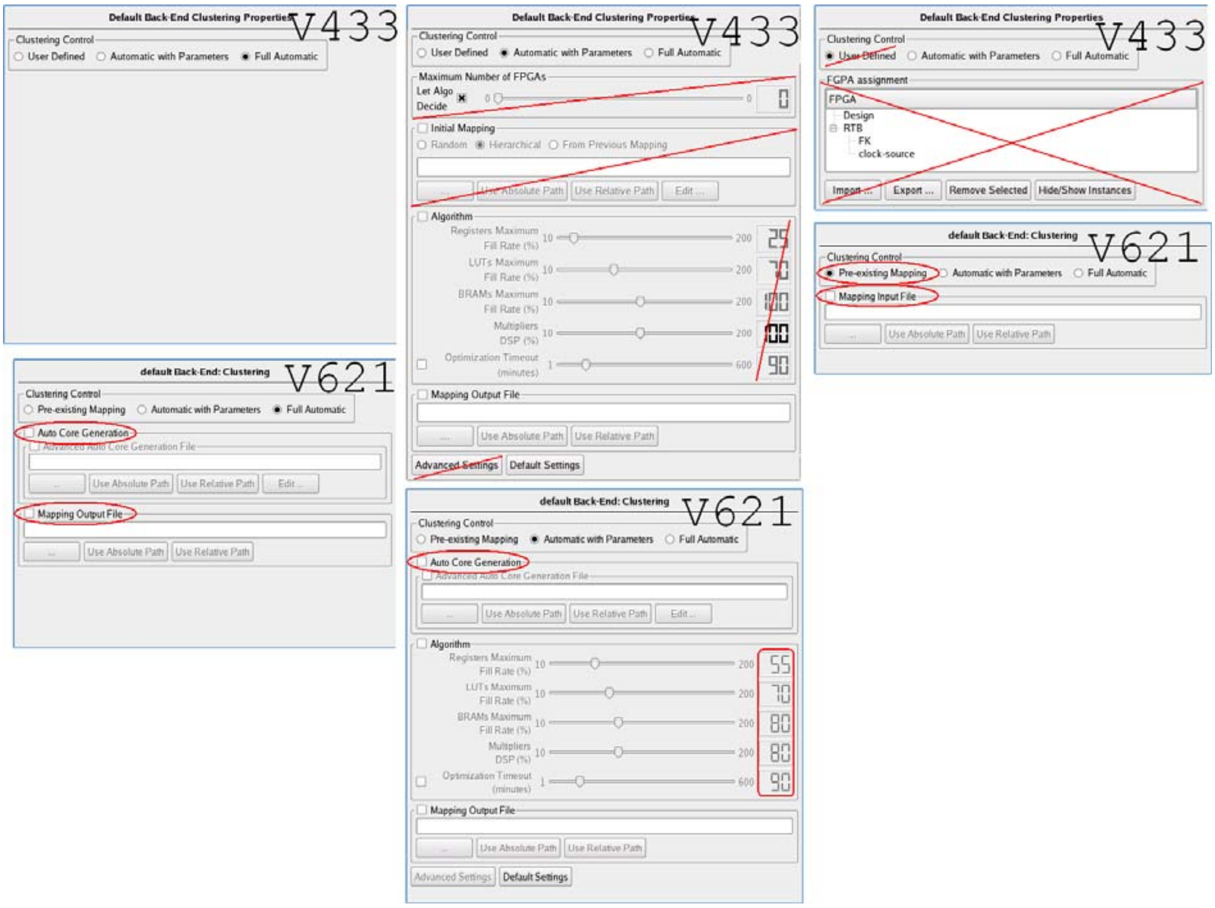

V433
Default Back-End Clustering Properties
Clustering Control
○ User Defined ○ Automatic with Parameters ● Full Automatic
Maximum Number of FPGAs
Let Algo Decide
Initial Mapping
○ Random ● Hierarchical ○ From Previous Mapping
Use Absolute Path
Use Relative Path
Edit ...
Algorithm
Registers Maximum Fill Rate (%)
LUTs Maximum Fill Rate (%)
BRAMs Maximum Fill Rate (%)
Multipliers DSP (%)
Optimization Timeout (minutes)
Mapping Output File
Advanced Settings
Default Settings
FPGA assignment
FPGA
Design
RTB
FK
clock-source
Import ...
Export ...
Remove Selected
Hide/Show Instances
V621
default Back-End: Clustering
○ Pre-existing Mapping ○ Automatic with Parameters ● Full Automatic
Auto Core Generation
Advanced Auto Core Generation File
Mapping Output File
Mapping Input File
25
70
100
100
90
55
70
80
80
90

Figure 4: *Back-End → Clustering* panel

- The **Automatic Core Generation** frame has been added to automatically create zCores for the design. This feature can be selected as an alternative to the automatic clustering feature which existed in V4_3_3. When **Automatic Core Generation** is selected, the settings in other frames are applicable to automatic partitioning instead of automatic clustering. This feature is described in the ***ZeBu-Server Release Note – V6_2_1B***.
- The **User-defined** clustering option has been replaced by **Pre-existing mapping**.

  Actually, it is not possible to create a user-defined clustering from scratch and/or to display it with an FPGA tree in V6_2_1B because of the multi-zCore architecture of the design, but once a mapping file has been created with automatic clustering (declared in **Mapping Output File** frame), this xml file can be modified manually for tuning purposes and declared in the **Mapping Input File** frame with **Pre-existing mapping** option.

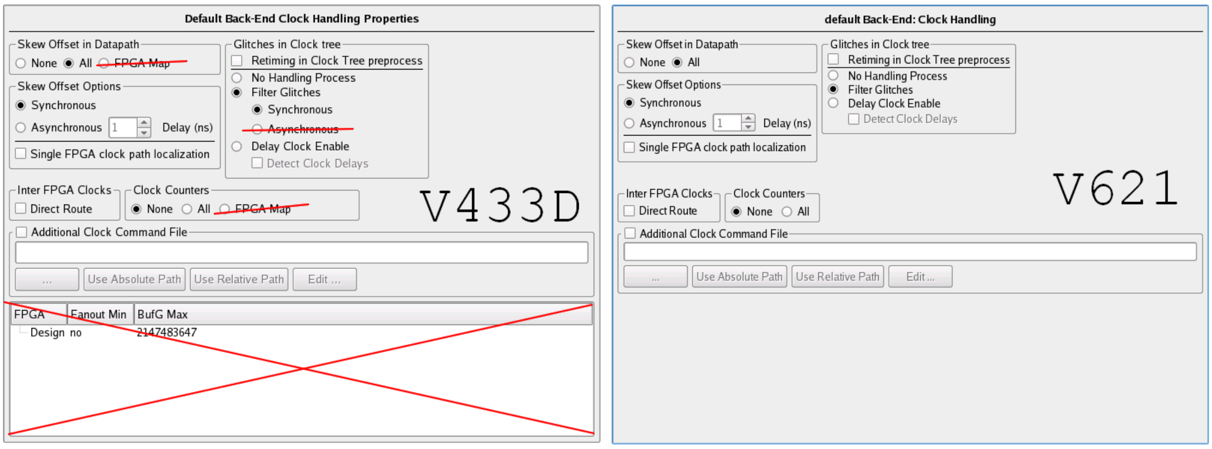
- When **Clustering control** → **User-defined** was selected in V4_3_3, the corresponding `defmapping` commands are no longer valid in V6_2_1B (because the system is different and because the clustering process has changed).

  With V6_2_1B, user must declare a mapping output file and use automatic clustering or automatic partitioning at first; switching to **Pre-existing Mapping** is possible once the compilation of the design has completed once.

- In **Automatic with Parameters**, the **Maximum Number of FPGAs** and **Initial Mapping** frames have been removed since they are no longer significant with the new compilation flow; the **Auto Core Generation** frame has been added. Note that the default values for filling rate settings have changed since the type of FPGAs is different.

- The additional tab for **Extra Commands** no longer exists in V6_2_1B. For information about the corresponding commands (manual memory mapping, ...), see Section 2.3.2.

2.1.3.6 Clock Handling



Default Back-End Clock Handling Properties

Skew Offset in Datapath: ○ None ● All ~~○ FPGA Map~~

Glitches in Clock tree: □ Retiming in Clock Tree preprocess; ○ No Handling Process; ● Filter Glitches (● Synchronous ~~○ Asynchronous~~); ○ Delay Clock Enable (□ Detect Clock Delays)

Skew Offset Options: ● Synchronous; ○ Asynchronous 1 Delay (ns); □ Single FPGA clock path localization

Inter FPGA Clocks: □ Direct Route

Clock Counters: ● None ○ All ~~○ FPGA Map~~

□ Additional Clock Command File

FPGA	Fanout Min	BufG Max
Design	no	2147483647

default Back-End: Clock Handling

Skew Offset in Datapath: ○ None ● All

Glitches in Clock tree: □ Retiming in Clock Tree preprocess; ○ No Handling Process; ● Filter Glitches; ○ Delay Clock Enable (□ Detect Clock Delays)

Skew Offset Options: ● Synchronous; ○ Asynchronous 1 Delay (ns); □ Single FPGA clock path localization

Inter FPGA Clocks: □ Direct Route

Clock Counters: ● None ○ All

□ Additional Clock Command File

Figure 5: ***Back-End*** **→** ***Clock Handling*** **panel**

The capability to modify the clock handling settings on a FPGA-by-FPGA basis is no longer available (**FPGA Map** option removed in **Skew Offset in Datapath** and **Clock Counters frames**).

The FPGA-based declaration for the maximum number of clock buffers for user clocks in design FPGAs (BUFGMAX) and for the minimum fanout is no longer available. The following commands can be added in the file declared in **System Parameters (zTopBuild)** → **Additional Command File** with V6_2_1 B_00:

- To declare a minimum fanout for all the FPGAs at once:

```
synchro fanout_min -value=<value>
```

- To declare the BUFGMAX for all FPGAs at once:

```
set_config -bufg <nb>
```

The **Filter Glitches** → **Asynchronous** option is no longer available in V6_2_1B: (the **Filter Glitches** option in V6_2_1B is equivalent to **Filter Glitches** → **Synchronous** option in V4_3_3).

If **Asynchronous** was selected in V4_3_3, you should select **Filter Glitches** → **Synchronous** in V6_2_1B.

2.1.3.7 Advanced

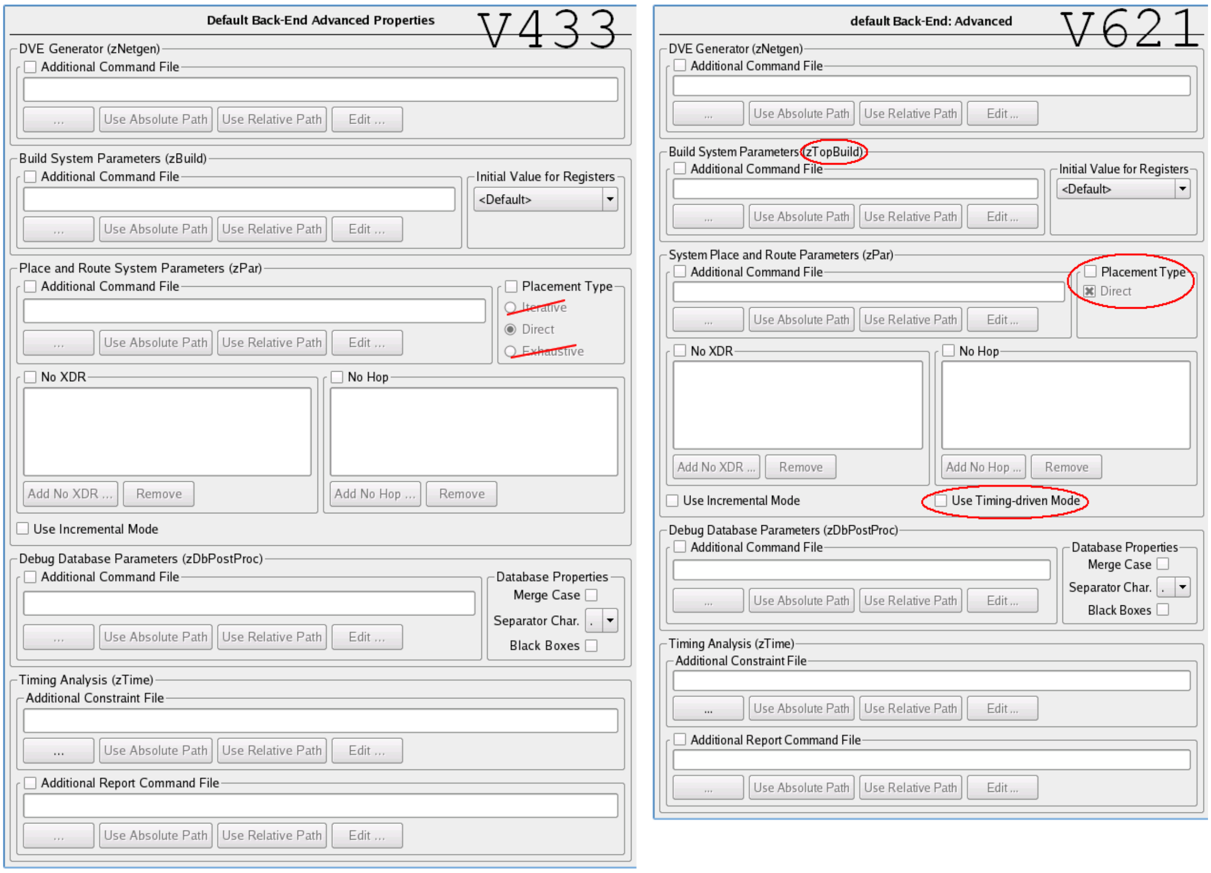

Default Back-End Advanced Properties
V433
DVE Generator (zNetgen)
□ Additional Command File
... Use Absolute Path Use Relative Path Edit ...
Build System Parameters (zBuild)
□ Additional Command File
Initial Value for Registers
<Default>
Place and Route System Parameters (zPar)
□ Additional Command File
□ Placement Type
○ ~~Iterative~~
● Direct
○ ~~Exhaustive~~
□ No XDR
□ No Hop
Add No XDR ... Remove
Add No Hop ... Remove
□ Use Incremental Mode
Debug Database Parameters (zDbPostProc)
□ Additional Command File
Database Properties
Merge Case □
Separator Char. .
Black Boxes □
Timing Analysis (zTime)
Additional Constraint File
□ Additional Report Command File
default Back-End: Advanced
V621
DVE Generator (zNetgen)
□ Additional Command File
Build System Parameters (zTopBuild)
□ Additional Command File
Initial Value for Registers
<Default>
System Place and Route Parameters (zPar)
□ Additional Command File
□ Placement Type
☒ Direct
□ No XDR
□ No Hop
Add No XDR ... Remove
Add No Hop ... Remove
□ Use Incremental Mode
□ Use Timing-driven Mode
Debug Database Parameters (zDbPostProc)
□ Additional Command File
Database Properties
Merge Case □
Separator Char. .
Black Boxes □
Timing Analysis (zTime)
Additional Constraint File
□ Additional Report Command File

Figure 6: *Back-End* → *Advanced* panel

In V6_2_1B, the **Build System Parameters (zTopBuild)** → **Additional Command File** is applicable for the system-level compiler only (**`zTopBuild`**).
There is no dedicated input field in **`zCui`** to declare a script for the zCore-level compiler (**`zCoreBuild`**): the specific script (`my_zcore_script.tcl`) has to be declared with its applicable zCore through a dedicated command in the system-level compiler script**:**

```
zcorebuild_script zcore <zcore_name> filename <my_zcore_script.tcl>
```

This command supports a wildcard character (*) as `<zcore_name>` if the script is applicable for all the zCores.

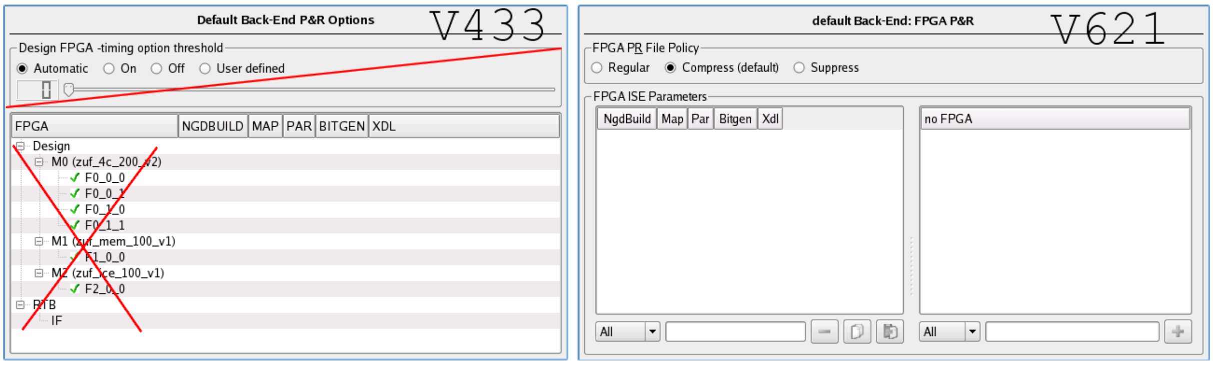
In the **Place and Route System Parameters** → **Placement Type** frame, only the **Direct** option is available in V6_2_1B. It was the default value in V4_3_3. Other values (**Iterative** and **Exhaustive**) have been deprecated because the communication architecture is very different in Zebu-Server with regard to ZeBu-XXL: the **Direct** option should work for any design.

The **Place and Route System Parameters → Use Timing-driven Mode** option is new in V6_2_1B (see ***ZeBu-Server Release Note*** for details). It is applicable when working on improving the performance of the design (once the design is already up-and-running).

All other frames are unchanged.

2.1.3.8 FPGA P&R

In the **FPGA P&R** panel, the **Design FPGA –timing option threshold** frame has been removed because the `-timing` option is no longer applicable with ISE 11 Place & Route.



Default Back-End P&R Options

Design FPGA -timing option threshold: ● Automatic ○ On ○ Off ○ User defined

default Back-End: FPGA P&R

FPGA PR File Policy: ○ Regular ● Compress (default) ○ Suppress

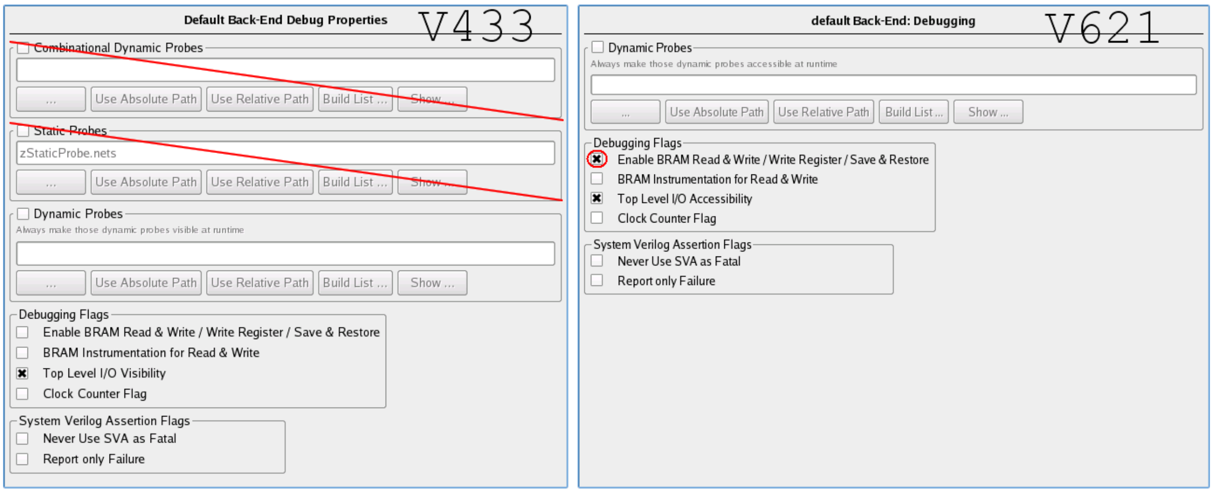
Figure 7: *Back-End → FPGA P&R* panel

The tree-like display has been removed since it did not fit the large number of FPGAs in ZeBu-Server. However, all the options set for ZeBu-XXL would be ignored since the FPGA type and the architecture of the systems are different.

The options in this panel cannot be set before at least one back-end compilation has been completed: the FPGA P&R step must have been run for all FPGAs, either successfully or not. This panel is described in Section 5.3 of the ***ZeBu-Server Compilation Manual*** (Rev b).

2.1.3.9 Debugging

Because the probe declaration has been moved to the **RTL-based Compilation Script**, as described in Section 2.1.2.1, the **Debugging** tab has been largely modified:



Default Back-End Debug Properties V433

~~□ Combinational Dynamic Probes~~

~~□ Static Probes~~

~~zStaticProbe.nets~~

□ Dynamic Probes
Always make those dynamic probes visible at runtime

Debugging Flags
□ Enable BRAM Read & Write / Write Register / Save & Restore
□ BRAM Instrumentation for Read & Write
☒ Top Level I/O Visibility
□ Clock Counter Flag

System Verilog Assertion Flags
□ Never Use SVA as Fatal
□ Report only Failure

default Back-End: Debugging V621

□ Dynamic Probes
Always make those dynamic probes accessible at runtime

Debugging Flags
☒ Enable BRAM Read & Write / Write Register / Save & Restore
□ BRAM Instrumentation for Read & Write
☒ Top Level I/O Accessibility
□ Clock Counter Flag

System Verilog Assertion Flags
□ Never Use SVA as Fatal
□ Report only Failure

Figure 8: *Back-End* → *Debugging* panel

The default value for the **Enable BRAM Read & Write / Write Register / Save & Restore** option has changed (it is set in V6_2_1B) but it is imported with its V4_3_3 value when loading the V4_3_3 compilation project in V6_2_1B.

Note that the declaration of signals in the **Dynamic Probes** frame is not actually used for compilation but these probes will automatically be selected at runtime for monitoring purposes, as it was in V4_3_3.

2.1.4 Preferences

The 3 panels for **Preferences** (**Job Scheduling**, **Compiler** and **Memories** panels) of **`zCui`** and their default values have not changed between V4_3_3 and V6_2_1.

Most recent modifications for V6_2_1B are described in the ***Zebu-Server Release Note – V6_2_1B***.

2.2 DVE File

2.2.1 No-longer supported declarations

- In case of a multi-process environment, the processes were declared in the DVE file for ZeBu-XXL. Note that such declaration is no longer supported and has been replaced by a declaration in the designFeatures file, as described in Section 3.1.5 of the ***ZeBu-Server Release Note for V6_2_1B***.

2.2.2 RTL-based declaration

It is possible to declare RTL hierarchical paths in the DVE file by adding the following line in it. When this parameter is present, the paths in the DVE file are automatically interpreted as RTL paths. Else, the paths in the DVE file are regarded as EDIF paths.

```
defparam rtlname = yes;
```

Where `yes` can be replaced by `true`.

Notes:

- RTL paths are not supported for elements such as `zceiClockPort`, `zClockPort` and `zIceClockPort`.
- RTL path expressions are Verilog compliant.
- All special characters are escaped.

Example:

The SRAM trace driver is instantiated with RTL paths which include some special characters:

```
defparam rtlname = yes;
SRAM_TRACE trace (
    .output_bin({
           top.mult1.r0,
           top.mult1.r1,
           top.mult1.\x[0] ,
           top.mult1.\z.1_
      })
);
```

2.2.3 Declaration of Smart Z-ICE / Direct ICE input clocks

In addition to the name of the Smart Z-ICE driver (SMART_ZICE_ZSE in V6_2_1B), most of the declarations for ZeBu-XXL Smart Z-ICE will be compatible for V6_2_1B software:

- The declaration of data connections is similar.
- The declaration of output clocks is similar.
- When declaring input clocks, a new type of clock port should be used: the `zceiClockPort` declaration of V4_3_3 DVE file should be changed into `zIceClockPort` for V6_2_1B.

2.3 zBuild Script

When the design was compiled for V4_3_3 software without using **zCui**, the content of the existing **zBuild** script cannot be used "as is": some commands are now replaced by items in the **zCui** graphical interface and others are declared either for the system-level compiler or for the zCore-level compiler in separate scripts.

2.3.1 zBuild commands changed to zCui items

The following **zBuild** commands have been replaced by settings through in the **zCui** graphical interface:

Table 1: zBuild commands changed to zCui items

zBuild command in V4_3_3 script	**zCui** corresponding item in V6_2_1B (panel → frame → item)	Described in the ZeBu-Server Compilation Manual (Rev b)
`synchro -is_clock_net`	**Clock Declaration**; **Type** = Internal Clock Path	§3.5.2
`synchro -is_clock_net -asynchronous`	*No equivalence for ZeBu-Server*	
`synchro -is_clock_port -mod`	*No equivalence for ZeBu-Server*	
`synchro -detect_clock_delays`	**Clock Handling → Glitches in Clock Tree → Delay Clock Enable → Detect Clock Delays**	§3.5.4.1
`synchro -no_clock_counters *`	**Clock Handling → Clock Counters → None** (default setting in V6_2_1B)	§3.5.4.1
`synchro -nodelay`	**Clock Handling → Skew Offset in Datapath** = None	
`enable instrument_bram_for_sw_rw *`	**Debugging → Enable BRAM Read & Write / Write Registers / Save & Restore** (selected by default in V6_2_1B)	
`source <system_conf>.tcl`	**Default Back End Configuration → Target Hardware Configuration File**	§4.1.1
`add -edif`	**Sources → Design → EDIF Sources → Add**	§3.1.1
`set_top <design_top_name>`	**Sources → Design → Top Name**	§3.2.4
`add -dve <dve_file_name>`	**Environment → DVE → User-defined**	§3.4.1.3
`cluster auto`	**Clustering → Full Automatic Clustering → Automatic with parameters**	§3.6.2
`disable dyn_probe_top`	**Debugging → Debugging Flags → Top Level I/O Accessibility**	§3.9.1
`set_top_dut <path>`	**Sources → Design → DUT Top Name**	
`reg init instance=* value=<0|1>`	**Advanced → Build System Parameters (zTopBuild) → Initial Value for Registers**	§ 3.7.1.1

`zBuild` command in V4_3_3 script	`zCui` corresponding item in V6_2_1B (panel → frame → item)	Described in the ZeBu-Server Compilation Manual (Rev b)
`disable dyn_probe_top`	**Debugging → Top Level IO Accessibility**	
`set work <work_dir>`	Automatically set from **Project → Project Working Directory**	

2.3.2 `zBuild` commands to be added in scripts declared in `zCui`

2.3.2.1 Syntax Recommendations

The syntax for the compilation commands has changed with regards to V4_3_3 but coherency has been improved:

- Mandatory parameters are simply listed after the command, for example:
```
command param1 param2.
```
- Optional parameters start with a hyphen, for example:
```
command param1 –option
```
- When an optional parameter needs a value, there is no = character. It is now a space character. For example:
```
command param -opt_value 1
```

For the command listed in the following tables, you should refer to the ZeBu-Server Compilation Manual for detailed syntax.

2.3.2.2 Equivalent Commands

Table 2: `zBuild` commands to be added in V6_2_1B scripts

`zBuild` command in V4_3_3 script	V6_2_1B command	`zCui` script in V6_2_1B (panel → frame → item)	Described in the ZeBu-Server Compilation Manual (Rev b)
`synchro -bufg_max <nb> *`	`set_config -bufg <n>`	**Advanced → Build System Parameters → Additional Command File**	
`tri_states –<type> <path>`	`tristate -<type> <path>`	**RTL-based compilation script** **Core Definition →**	§3.7.3

`force undriven value=<value>`	`force undriven`
	`–value <value>`
	`[-reg_init 0|1]`
	`[-verbose yes|no]`
		Core Definition → Netlist Edition File	§3.7.4.2

`force pin`	`force assign -pin`	**Core Definition → Netlist Edition File**	
`force net`	`force assign –net`	**Core Definition → Netlist Edition File**	§3.7.4.5

`zBuild` command in V4_3_3 script	V6_2_1B command	`zCui` script in V6_2_1B (panel → frame → item)	Described in the ZeBu-Server Compilation Manual (Rev b)
`force assign`	`force assign`	**Core Definition → Netlist Edition File**	§3.7.4
`force default verbose=<yes|no>`	`force default –globalVerbose <yes|no>`	**Core Definition → Netlist Edition File**	§3.7.4.3
`blackbox define module=<mod>`	`blackbox define <mod>`	**Core Definition → Netlist Edition File**	§3.7.6.1
`blackbox remove module=<mod>`	`blackbox remove <mod>`	**Core Definition → Netlist Edition File**	§3.7.6.2
`blackbox drive module=<mod> port=<port> value=1|0`	`blackbox drive <mod> -port <port> - value {0|1|REG}`	**Core Definition → Netlist Edition File**	§3.7.6.3
`blackbox ignore module=<mod>`	`blackbox ignore <mod>`	**Core Definition → Netlist Edition File**	§3.7.6.1

2.3.2.3 Automatic Clustering commands

When no zCore is declared by user in V6_2_1B, the existing `cluster` commands of the V4_3_3 **`zBuild`** script can be added in a **`zCoreBuild`** script to control automatic clustering in the only zCore of the design. The V4_3_3 syntax can be used in this script. This zCoreBuild script has to be declared as described in Section 2.1.3.7.

2.3.2.4 Defmapping Commands

Note that the `defmapping` commands in V6_2_1B are applicable for a given zCore and cannot be pasted from the ZeBu-XXL script because the ZeBu-Server architecture is very different.
With V6_2_1B, it is mandatory to start compiling with automatic clustering and then switch to **Pre-Existing Mapping** as described in Section 2.1.3.5. The syntax for FPGA identification has changed from F*m*_*c*_*s* to U*x*.M*y*.F*z*.

Table 3: `defmapping` commands

`zBuild` command in V4_3_3 script	V621 command	`zCui` script (panel → frame → item)	Described in the Compilation Manual Rev B
`defmapping –clock_source`	*No Equivalence for ZSE*		
`defmapping <inst_path> FK`	*No Equivalence for ZSE*		
`defmapping <mem_path> <FPGA>`		**`zCoreBuild`** script	
`defmapping [–mem_type <type>] <mem_path> [FPGA]`		**`zCoreBuild`** script	
`defmapping –mem <mem_path> <mem_id>`		**`zCoreBuild`** script	

2.3.2.5 Deprecated commands

The following commands were described for ZeBu-XXL (in the ***ZeBu-UF Compilation Manual*** - Rev b). They are outdated and should be removed from compilation scripts in V6_2_1B:

- `enable fpga_async_path_wire`
- `set zcf_path <path>`
- `set vector_format <char>`
- `enable top_inout_orientation`
- `force default tcl_file_name`
- `enable quit_before_uniquify`
- `set vectorize verbose`
- `enable adjust_bin_top_inout`
- `force default create_tcl`

If you encounter any issue due to scripts which had these commands, you should contact EVE support.

2.4 zPar script

The System Place and Route is very different between ZeBu-XXL and Zebu-Server. It is strongly recommended to first try the default settings of V6_2_1B when migrating from ZeBu-XXL environment.

The following commands are no longer applicable for ZeBu-Server and must not be used in an **Additional Command File** declared in the **Advanced** → **System Place and Route (zPar)** frame:

- `System routeLowSkew <wirename>`
- `System noXilinxDdr`
- `System noreroute`
- `System xdrCtrlMaxFanout`
- `System sdrGates <buffer|reg|latch> <buffer|reg|latch>`
- `System simpleCkRouting`
- `System simpleCkRouting`
- `System disableFKpairs`

2.5 designFeatures File

- The module identifiers have been modified for ZeBu-Server since multi-unit environments are supported. The designFeatures declaration which start with `$Bn.*` for Zebu-XXL must be modified into `$Ux.My.*` when migrating to ZeBu-Server (for compatible declarations).

- The multi-process feature in V6_2_1B is different of what it was for ZeBu-XXL, as described in Section 2.2.1.

  The main difference is that the number of processes and the process name must be declared in the `designFeatures` file instead of the DVE file. For example:

  ```
  $nbProcess = 2;
  $process_0 = "default_process";
  $process_1 = "jtag_process";
  ```

3 EVE Contacts

For product support, contact: support@eve-team.com.

For general information, visit our company web-site: http://www.eve-team.com

Europe Headquarters	EVE SA 2-bis, Voie La Cardon Parc Gutenberg, Bâtiment B 91120 Palaiseau FRANCE Tel: +33-1-64 53 27 30
US Headquarters	EVE USA, Inc. 2290 N. First Street, Suite 304 San Jose, CA 95054 USA Tel: 1-888-7EveUSA (+1-888-738-3872)
Japan Headquarters	Nihon EVE KK KAKiYA Building 4F 2-7-17, Shin-Yokohama Kohoku-ku, Yokohama-shi, Kanagawa 222-0033 JAPAN Tel: +81-45-470-7811
Korea Headquarters	EVE Korea, Inc. 804 Kofomo Tower, 16-3, Sunae-Dong, Bundang-Gu, Sungnam City, Kyunggi-Do, 463-825, KOREA Tel: +82-31-719-8115
India Headquarters	EVE Design Automation Pvt. Ltd. #143, First Floor, Raheja Arcade, 80 Ft. Road, 5th Block, Koramangala Bangalore - 560 095 Karnataka INDIA Tel: +91-80-41460680/30202343
Taiwan Headquarters	EVE-Taiwan Branch 5F-2, No. 229, Fuxing 2nd Rd. Zhubei City, Hsinchu County 30271, TAIWAN (R.O.C.) Tel: +886-(3)-657-7626