

Migrating from BIOS 4.9 to BIOS 5.x

This topic discusses several common questions when updating to DSP/BIOS 5.x using the CDB2TCF utility.

Contents

General statements:

Files generated by the CDB conversion process
Additional details about the conversion process can be found at

Q: When I try to open the newly converted .TCF file, why do I see a big error screen about heaps?

Q: When I try to open the newly converted .TCF file, why do I see a big error screen about reference constraint violation?

Q: I am using C64x DSP family. Why does the CDB2TCF utility fail?

Q: I am using C55x DSP family. When I try to open the newly converted TCF file, why do I see a big error screen that shows "Cannot call method 'create' of undefined"?

Q: I am using C55x DSP family. After verifying the TCF file was properly converted (no errors when opening it), why does the project still have errors when I try to build it?

Q: I am using C28x DSP family. Why has the M1SARAM memory disappeared from my newly generated .TCF file?

Q: I am using C54x DSP family. The conversion process is fine, but why I keep getting an error for the value MEM.BIOSNORPTBSEG in my generated .TCF file?

Q: How do I compare before and after cdb conversion to be sure it's right?

General statements:

It is highly recommended BIOS to be updated to the latest release - it contains several bug fixes to the conversion process.

The recommended version for legacy DSPs is 5.31.x and for all newer ISAs is 5.32.x. All are available at the Update Advisor webpage.

If CCS 3.1 is updated to BIOS 5.x, the conversion from any previous BIOS release is automatic (whenever the old CDB file is opened).

If CCS 3.3 is being used the conversion from BIOS 4.90 is also automatic, since out-of-the-box BIOS 5.x contains seed files used by version 4.90.

However, if the original CDB file was created using an older BIOS release then an additional step is required for CCS 3.3. The file **update.zip** located at %BIOS_INSTALL_DIR%\packages\ti\bios\config\update must be unzipped before opening the CDB file. This way the original seed files for all previous DSP/BIOS releases will be available.

Files generated by the CDB conversion process

The process of converting BIOS 4.90 CDB files to BIOS 5.x TCF files generates two files: one TCF file and one updated CDB file (usually with smaller size). It also renames the original CDB file by appending the date and time information:

For example: if the original CDB file is <example.cdb> it will be renamed to <example.cdb.Thu_Jul_17_16.07.07_2008>

This new CDB file generated only contains a partial BIOS configuration and cannot be used independently from the TCF.

Therefore, if the original CDB file was successfully converted the first time you opened one of the example projects, keep in mind that other projects may still refer to the same <example.cdb> file. In order to avoid this issue, when you open an example project say 'no' to the Project Update message box and manually replace the <example.cdb> file with the <example.tcf> file.

Additional details about the conversion process can be found at

- the <Setup Guide.html> located at the DSP/BIOS installation directory %BIOS_INSTALL_DIR%\SetupGuide.html
- Ti's knowledge base article located [here \(http://www-k.ext.ti.com/SRVs/CGI-BIN/WEBCEG1.EXE/?St=180,E=0000000000011290246,K=3716,Sxi=1,Case=obj\(52425\)\)](http://www-k.ext.ti.com/SRVs/CGI-BIN/WEBCEG1.EXE/?St=180,E=0000000000011290246,K=3716,Sxi=1,Case=obj(52425))

Q: When I try to open the newly converted .TCF file, why do I see a big error screen about heaps?

If a BIOS 4.90 project has configured heaps, BIOS 5.x may fail to properly enable them after converting the project and spawns the following error message:

```
js: ".\c64p.cdb", line 11: Heaps are enabled, but the segment for DSP/BIOS Objects (MEM.BIOSOBJSEG) is not set correctly.
Please create a heap in one of your data segments and set MEM.BIOSOBJSEG to it (and so on)
```

This error happens because early releases of BIOS 5.x do not properly allocate the heap as a default - this was fixed in BIOS releases 5.31.03 and newer.

In order to properly allocate the heap, at least one of the memory segments must have the box "Create a heap in this memory" checked (see this [screen capture](#)):

The next step is to choose which memory segment that contains a heap will be used. Right-click on MEM - Memory Section manager and select Properties. Select a memory segment for both BIOS objects and malloc() - these can't be MEM_NULL (see this [screen capture](#)):

Technically speaking, prior versions of the configuration tool allowed for an inconsistent memory heap configuration, enabling heaps when no heap memory space was actually configured. This caused problems in applications since malloc() and MEM_alloc() failed even if BIOS enabled dynamic memory allocation. The error message above is spawned because BIOS checks to make sure that configurations that have heaps also have a valid memory segment assigned to them. In other words, the parameters bios.MEM.MALLOCSEG and bios.MEM.SEGZERO are checked if they are set to a valid heap instead of "MEM_NULL".

Q: When I try to open the newly converted .TCF file, why do I see a big error screen about reference constraint violation?

CDB2TCF tool may assign invalid memory segments to unused objects. If any modules are disabled in the CDB file (like RTDX, CLK, BUF, etc.) but happen to be assigned to an invalid memory segment, BIOS 4.9 does not throw any error and the file builds ok. When the conversion process takes place, BIOS 5.31 may blindly assign the same invalid memory segments to these objects, causing the error. An example follows below:

```
js: ".\dsk6455.tcf", line 206: Reference constraint violation: IRAM is an illegal value for BUF0BJMEMSEG "./", line 4
js: ".\dsk6455.tcf", line 246: Reference constraint violation: IRAM is an illegal value for CLK0BJMEMSEG "./", line 4
js: ".\dsk6455.tcf", line 252: Reference constraint violation: IRAM is an illegal value for RTDX.DATAMEMSEG "./", line 4
Correct any script errors in C:/CCStudio_v3.3/My Projects/dsk6455/dsk6455.tcf using the Configuration
Tool or a text editor, if necessary.
TConf initialization arguments: -e environment['config.importPath']='C:/CCStudio_v3.3/bios_5_31_08/packages'
-e environment['config.scriptName']='dsk6455.tcf'
```

The workaround is to search for the modules and reassign them to valid memory segments. In the example above, supposing you have a valid memory segment named "SDRAM", the changes would be:

Replace `<syntaxhighlight lang="c">bios.CLK.OBJMEMSEG = prog.get("IRAM");</syntaxhighlight>` with `<syntaxhighlight lang="c">bios.CLK.OBJMEMSEG = prog.get("SDRAM");</syntaxhighlight>`

Replace `<syntaxhighlight lang="c">bios.RTDX.RTDXDATASEG = prog.get("IRAM");</syntaxhighlight>` with `<syntaxhighlight lang="c">bios.RTDX.RTDXDATASEG = prog.get("SDRAM");</syntaxhighlight>`

Q: I am using C64x DSP family. Why does the CDB2TCF utility fail?

If you are using CCS 3.2, its installation does not supply the seed files used by the CDB2TCF tool to properly convert the CDB file. The error below is shown:

```
Running "C:\CCStudio_v3.2\MyProjects\copy_sim64xx\cdb2tcf.bat"
Please wait . . .
Warning: BIOS_INSTALL_DIR differs from the BIOS installation from which cdb2tcf
is executed.
BIOS INSTALL DIR: c:\ccstudio_v3.2\bios_5_30_00_06/
cdb2tcf path: C:/CCStudio_v3.2/bios_5_30_00_06/
usage: cdb2tcf [-h] [-d] [-i <customImport>.tci <customSeed>.cdb] [-l <logfile>]
[-a <verbosity level>] <application>.cdb
Conversion FAILED. This is the command execution log:
cdb2tcf command FAILED.
Press any key to continue . . .
```

One possible workaround is to copy the seed files supplied with CCS 3.1 (if installed) from C:\CCStudio_v3.1\C6000\bios\include.

Another workaround is update to BIOS release 5.30 and newer.

Q: I am using C55x DSP family. When I try to open the newly converted TCF file, why do I see a big error screen that shows "Cannot call method 'create' of undefined"?

BIOS releases 5.31.02 and older have a bug in the file `<utils.tci>` that searches for the device names using uppercase letters only, thus preventing the converted .TCF files to properly find them. The following error is spawned:

```
js: "C:/CCStudio_v3.3/bios_5_30/packages/ti/platforms/generic/Platform.tcp", line 57: org.mozilla.javascript.UniqueTag
"C:/CCStudio_v3.3/bios_5_30/xdctools/include/utils.tci", line 562
"C:/CCStudio_v3.3/bios_5_30/xdctools/include/utils.tci", line 791
"/ledprd.tcf", line 10
".<stdin>", line 4
js: "C:/CCStudio_v3.3/bios_5_30/xdctools/include/utils.tci", line 834: exception from uncaught JavaScript throw: TypeError:
Cannot call method "create" of undefined (C:/CCStudio_v3.3/bios_5_30/xdctools/include/utils.tci#834)
"/ledprd.tcf", line 10
".<stdin>", line 4
Correct any script errors in C:/CCStudio_v3.3/boards/evm5509a/examples/ledprd/ledprd.tcf
using the Configuration Tool or a text editor, if necessary.
TConf initialization arguments:
-e environment['config.importPath']='c:/ccstudio_v3.3/bios_5_30/packages;C:/CCStudio_v3.3/bios_5_30/packages;' -e
environment['config.scriptName']='ledprd.tcf'
```

The workaround is to update the BIOS release to 5.31.03 and newer.

In rare cases a possible workaround is to find the line below (or similar) in the generated .TCF file:

`<syntaxhighlight lang="c">`

```
params.deviceName = "5509a";
```

`</syntaxhighlight>`

And replace it with:

`<syntaxhighlight lang="c">`

```
params.deviceName = "5509A";
```

`</syntaxhighlight>`

Q: I am using C55x DSP family. After verifying the TCF file was properly converted (no errors when opening it), why does the project still have errors when I try to build it?

After converting to BIOS 5.x a C55x project that uses SMALL memory model, linker may report that certain files are missing:

```
error: cannot find file "lnkrtdx.a55"
error: cannot find file "drivers.a55"
error: cannot find file "sioboth.a55"
error: cannot find file "bios5510.a55"
error: cannot find file "bios.a55"
error: cannot find file "rtdxsim.lib"
```

BIOS 5.x does not support small memory model for C55x family of DSPs, therefore if the original BIOS 4.90 project used this memory model, the lines below are automatically added to the generated .tcf file and the last one must be uncommented:

<syntaxhighlight lang="c">

```
/* SMALL memory model is not supported. To build for LARGE model, uncomment
 * the following statement and add '-ml' to your compiler options.
 */
/* bios.GBL.MEMORYMODEL = "LARGE";*/
```

</syntaxhighlight>

Another workaround is to change the memory model using the graphical configuration tool. Go to System-> Global Settings. Right-click on Properties and then select either LARGE or HUGE in the Memory Model option - see this [screen capture](#).

In both cases, do not forget to set the compiler option -ml to the menu Project -> Build Options -> tab Compiler -> section Advanced - see this [screen capture](#).

Q: I am using C28x DSP family. Why has the M1SARAM memory disappeared from my newly generated .TCF file?

CDB2TCF fails to properly define the M1SARAM memory segment, therefore the error.

A workaround is to open the .TCF file in a text editor and manually add the M1SARAM section before any references to it in the file. More specifically, add the following lines just before the line `/* enabling DSP/BIOS components */`: <syntaxhighlight lang="javascript"> bios.M1SARAM = bios.MEM.create("M1SARAM"); bios.M1SARAM.base = 0x400; bios.M1SARAM.len = 0x400; </syntaxhighlight>

Make sure the previously defined section MoSARAM does not overlap the newly created one - typically MoSARAM ranges from **0x000000** to **0x0003FF**.

When a 28x TCF file is created from scratch, MoSARAM and M1SARAM are defined as a single block called MSARAM.

Q: I am using C54x DSP family. The conversion process is fine, but why I keep getting an error for the value MEM.BIOSNORPTBSEG in my generated .TCF file?

If the error you are getting is similar to this:

```
js: "./DSP_C54_BIOS.tcf", line 225: Reference constraint violation: IPROG is an illegal value for MEM.BIOSNORPTBSEG
  ./<stdin>", line 4
Correct any script errors in C:/CCStudio_v3.3/MyProjects/test/DSP_C54_BIOS.tcf using the Configuration Tool or a text editor, if necessary.
TConf initialization arguments:
-e environment['config.importPath']='C:/CCStudio_v3.3/bios_5_31_02/packages' -e environment['config.scriptName']='DSP_C54_BIOS.tcf'
```

Your .TCF file is probably allocating the memory segment IPROG in extended or external memory.

In order to use extended memory addressing in C54x DSPs, a specific memory section called `.bios:.norptb` must be created and allocated inside the overlay page (internal memory space) - this is described in the document [SPRA599A](http://focus.ti.com/general/docs/techdocsabstract.tsp?abstractName=spra599a) (<http://focus.ti.com/general/docs/techdocsabstract.tsp?abstractName=spra599a>).

In BIOS 4.90 this was done using an external linker command file .CMD created by the user. However, in BIOS 5.x the directive MEM.BIOSNORPTBSEG automatically creates this memory section and allocates it to the default internal memory segment IPROG.

Therefore, if your original CDB file contains a memory segment called IPROG but allocates it to extended memory, the conflict will be caught by BIOS 5.x.

Workarounds to fix this:

- Comment out lines that re-allocate the internal IPROG memory segment (see below). By default BIOS will create a memory segment IPROG;
- Redefine this memory segment to a lower address inside the overlay page;
- Recreate the memory segment on these lines to something else other than IPROG - it can be created using the directive `bios.MEM.create()` seen above

In the .TCF script these lines would be something similar to: <syntaxhighlight lang="javascript"> bios.IPROG.comment = "Program memory"; bios.IPROG.base = 0x1000; /* bad memory address - should be located in internal memory */ bios.IPROG.len = 0x1000; </syntaxhighlight>

Q: How do I compare before and after cdb conversion to be sure it's right?

Try this.

- `c:\ccstudio_v3.3\bios_5_31_08\dxctools\cdb2tcf.exe tci_test.cdb`

This produces a Tconf script as indicated above.

Note - if this fails it is probably because the base seed file (the original baseline of your original cdb file) is not there - follow the instructions in this article re update.zip

Note - you may need to set your BIOS_INSTALL_DIR first e.g. `set BIOS_INSTALL_DIR=c:/ccstudio_v3.3/bios_5_31_08`

```
▪ tconf -Dconfig.importPath="c:\ccstudio_v3.3\bios_5_31_08/packages" tci_test.tcf
```

This will generate the updated cdb based on your newly created TCF file.

```
▪ c:\CCStudio_v3.3\bios_5_31_08\xdctools\cdbcmp.exe tci_test.cdb.Fri_Oct_10_10.45.38_2008 tci_test.cdb
```

The cdbcmp utility compares 2 cdb files. Unfortunately cdbcmp is pretty noisy so look through it for important differences.

What differences are *not* important?

Assertions. Example: -

```
// value changed:
// old HWI_INT4.Use Dispatcher = 0
// new HWI_INT4.Use Dispatcher = 1
assert.add('prog.module("HWI").instance("HWI_INT4").useDispatcher == 1');
```

Order properties. Your code should not be relying on the order of additions of tasks etc in the config. It should be using priorities etc effecticely. For the most part the .order property can be ignored.

```
// value changed:
// old my_SWI.Order = 7
// new my_SWI.Order = 0
prog.module("SWI").instance("my_SWI").order = 0;
```

CSL graphical component deletions. As indicated in this topic, Chip Support Library config is no longer supported. Hence the cdbcmp tool will show stuff like: -

```
// type EMIF deleted
prog.module("EMIF").destroy();
```

This should help eliminate the unimportant stuff so you can look for potentially important differences.

<pre> {{ 1. switchcategory:MultiCore= ▪ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum ▪ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum Please post only comments related to the article Migrating from BIOS 4.9 to BIOS 5.x here. </pre>	<p>Keystone=</p> <ul style="list-style-type: none"> ▪ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum ▪ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article Migrating from BIOS 4.9 to BIOS 5.x here.</p>	<p>C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article Migrating from BIOS 4.9 to BIOS 5.x here.</p>	<p>DaVinci=For technical support on the DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article Migrating from BIOS 4.9 to BIOS 5.x here.</p>	<p>MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article Migrating from BIOS 4.9 to BIOS 5.x here.</p>	<p>OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Migrating from BIOS 4.9 to BIOS 5.x here.</p>	<p>OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Migrating from BIOS 4.9 to BIOS 5.x here.</p>	<p>MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article Migrating from BIOS 4.9 to BIOS 5.x here.</p>	<p>For technical support please post your questions at http://e2e.ti.com. Please post on comments about article Migrating from BIOS 4.9 to BIOS 5.x here.</p>
	<pre> }} </pre>							

Links



- [Amplifiers & Linear Audio](#)
- [Broadband RF/IF & Digital Radio](#)
- [Clocks & Timers](#)
- [Data Converters](#)

- [DLP & MEMS High-Reliability Interface](#)
- [Logic Power Management](#)

Processors

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

Switches & Multiplexers

- [Temperature Sensors & Control ICs](#)
- [Wireless Connectivity](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=Migrating_from_BIOS_4.9_to_BIOS_5.x&oldid=181377"

This page was last edited on 11 July 2014, at 11:13.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.