

## 2.12.2 Flash Tools

Texas Instruments provides the following tools for flash:

- Code Composer Studio (CCS) - the development environment with integrated flash plugin
- F021 Flash API Library - a set of software peripheral functions to erase/program flash  
Please refer to the *C2000 F021 Flash Application Programming Interface (API) Reference Guide* for more information.
- UniFlash - standalone tool to erase/program/verify the flash content through JTAG. No CCS is required.
- Linker ECC generation - to generate the Flash ECC from the Flash data and to introduce errors in Flash/ECC space. Please refer to the *TMS320C28x Assembly Language Tools User's Guide (SPRU513)* for more information.
- CCS On-Chip Flash Plugin and UniFlash tools are developed for a fully Flash-embedded application, which should be a standalone program that has all the initialized sections linked to flash. Any code that needs to be executed from RAM should be copied from Flash at run time.

### 8.5.10 Configuring Error Correcting Code (ECC) with the Linker

Error Correcting Codes (ECC) can be generated and placed in separate sections through the linker command file. ECC uses extra bits to allow errors to be detected and/or corrected by a device. To enable ECC generation, you must include `--ecc=on` as a linker option on the command line. By default ECC generation is off, even if the ECC directive and ECC specifiers are used in the linker command file. This allows you to fully configure ECC in the linker command file while still being able to quickly turn the code generation on and off via the command line.

The ECC support provided by the linker is compatible with the ECC support in TI Flash memory on various TI devices. TI Flash memory uses a modified Hamming(72,64) code, which uses 8 parity bits for every 64 bits. Check the documentation for your Flash memory to see if ECC is supported. (ECC for read-write memory is handled completely in hardware at run time.)

You can control the details of ECC generation using the ECC specifier in the memory map ([Section 8.5.10.1](#)) and the ECC directive ([Section 8.5.10.2](#)).

See [Section 8.4.12](#) for command-line options that introduce bit errors into code that has a corresponding ECC section or into the ECC parity bits themselves. Use these options to test ECC error handling code.

ECC can be generated during linking. The ECC data is included in the resulting object file, alongside code and data, as a data section located at the appropriate address. No extra ECC generation step is required after compilation, and the ECC can be uploaded to the device along with everything else.

