



Symbian OS has its own platform-independent build file format (used by bld.inf) to specify how a program is built. The bldmake tool processes the bld.inf component description file to generate a batch file, abld.bat.

? **bldmake bldfiles** generates abld.bat

? **bldmake clean** removes all files generated by bldmake bldfiles

? **bldmake inf** displays the basic bld.inf syntax

? **bldmake plat** displays a list of supported build platforms.

Build platforms represent the various target platforms (and thus binary formats) supported by Symbian OS. For Symbian OS v9, the most commonly used build platforms are:

? **WINS32**, which creates x86-format binaries for running code on the Windows emulator

? **GCCE or ARMV5**, which create binaries to run on phone hardware, built with the GCCE and RVCT compilers respectively. When one of the platforms listed is specified as an argument to the abld.bat command, the makefile for that platform is generated and executed.

The abld.bat command can be invoked from the command line with various arguments. The most commonly used are as follows:

? **abld build** combines a number of other arguments (export, makefile, library, resource, target and final) in turn to build the components specified as MMP files under the PRJ_MMPFILES specifier in the bld.inf file. Likewise, abld test build builds those components specified under PRJ_TESTMMPFILES in bld.inf

? **abld makefile** creates the makefiles for each project specified in bld.inf, using the Symbian OS makmake tool. The makefiles are then used by abld to carry out the various stages of building the component. This command is called each time a component is built, and the makefiles are always re-created, regardless of whether the corresponding MMP files have been changed or not since their last creation

? **abld freeze** freezes new DLL exports into .def files. For more detail, see Chapter 10 which describes the use of the IMPORT_C macro to specify DLL exports and the roles of the module definition file and import library

? **abld clean** erases all the files created by a corresponding abld target command (all the intermediate files created during compilation and all the executables and import libraries created by the linker)

? **abld reallyclean** does what abld clean does, and also removes files exported by abld export and makefiles generated by abld makefile, or the abld test equivalents.

Example:

Step 1: Go to the directory where bld.inf file is present. Open the command prompt

Example: C:\Symbian\9.2\S60_3rd_FP1\Examples\Base\ArraysAndLists\DynamicArrays>**bldmake bldfiles**

How_to_build_an_application_in_Command_Prompt

note: if there is no error command prompt will return back.if you j=just check the directory a new file is generated named ABLD.BAT (windows bat file)

Step 2:

```
C:\Symbian\9.2\S60_3rd_FP1\Examples\Base\ArraysAndLists\DynamicArrays> abld build gccce urel
```

Thus we have succesfully compiled the application if it retruns no error .

Bold text