Contemporary SoC design involves the proper selection of cores from a reference platform. Such selection implies the design exploration of alternative CPUs, which requires the generation of binary code for each possible target. However, the embedded computing market shows a broad spectrum of instruction-set architectures, ranging from micro-controllers to RISCs and ASIPs. As a consequence, binary utilities cannot always rely on pre-existent tools within standard packages. Besides, the task of manually retargeting every binary utility is not acceptable under time-to-market pressure. This paper describes a technique for the automatic generation of binary utilities from an abstract model of the target CPU, which can be synthesized from an arbitrary ADL. The technique is based upon two key mechanisms: model provision for tool generation (at the front-end) and automatic library modification (at the backend). To illustrate the technique's automation effectiveness, we describe the generation of assemblers, linkers and disassemblers. We have successfully compared the files produced by the generated tools to those produced by conventional tools. Moreover, to give proper evidence of retargetability, we present results for MIPS, SPARC, PowerPC and i8051.