Bound-T timing analysis tool

Technical Note

CDB from SDCC as input to Bound-T

TR-TN-CDB-001
Version 1
2009-05-19
Tidorum Ltd.

Tidorum Ltd
www.tidorum.fi
Tiirasaarentie 32
FI-00200 Helsinki
Finland

This document was written and is maintained by Niklas Holsti at Tidorum Ltd.

Document reference: TR-TN-CDB-001
Document issue: Version 1
Document issue date: 2009-05-19
Bound-T version: Various, depending on target processor.
Web location: *http://www.bound-t.com/tech_notes/tn-cdb.pdf*

Credits:
This document was created with the free OpenOffice.org software, *http://www.openoffice.org/*.

Relevant Bound-T source-file versions:

File	*Revision*
formats-sdcc.adb	1.1
formats-sdcc.ads	1.2
formats-sdcc-cdb.adb	1.5
formats-sdcc-cdb.ads	1.3

The information in this document is believed to be complete and accurate when the document is issued. However, Tidorum Ltd. reserves the right to make future changes in the technical specifications of the product Bound-T described here. For the most recent version of this document, please refer to the web address *http://www.bound-t.com/tech_notes/tn-cdb.pdf*.

If you have comments or questions on this document or the product, they are welcome via electronic mail to the address *info@tidorum.fi*, or via telephone, fax or ordinary mail to the address given below.

Please note that our office is located in the time-zone GMT + 2 hours (+3 hours in the summer) and office hours are 9:00 -16:00 local time.

Cordially,

Tidorum Ltd.

Telephone: +358 (0) 40 563 9186
Fax: +358 (0) 42 563 9186
Web: *http://www.tidorum.fi/*
http://www.bound-t.com/
Mail: *info@tidorum.fi*

Post: Tiirasaarentie 32
FI-00200 HELSINKI
Finland

Credits

The Bound-T tool was first developed by Space Systems Finland Ltd (*http://www.ssf.fi*) with support from the European Space Agency (ESA/ESTEC). Free software has played an important role; we are grateful to Ada Core Technology for the Gnat compiler, to William Pugh and his group at the University of Maryland for the *Omega* system, to Michel Berkelaar for the *lp-solve* program, to Mats Weber and EPFL-DI-LGL for Ada component libraries, and to Ted Dennison for the *OpenToken* package. Call-graphs and flow-graphs from Bound-T are displayed with the *dot* tool from AT&T Bell Laboratories. Some versions of Bound-T emit XML data with the *XML_EZ_Out* package written by Marc Criley at McCae Technologies.

Contents

Index of Tables

1 INTRODUCTION

1.1 Purpose and scope

Bound-T is a tool for computing bounds on the worst-case execution time of real-time programs; see reference [1]. Bound-T applies static analysis to the machine-code program in its compiled, linked and executable form. Bound-T must therefore start by reading in the machine-code program from a file, for example a file in the ELF format. Bound-T can read and understand several program-file formats. In addition to the basic machine code (memory load image) a program file usually also contains symbolic debugging information for the program (and this is usually the complex part of the format).

There are different versions of Bound-T for different target processors. Each version supports a set of program formats that are commonly used with that target processor and its cross-compilers. The Bound-T Application Note for each target explains which formats are supported. Some formats combine the machine code and symbolic debugging information in the same file; others use separate files for the two sets of data.

This Technical Note supplements the Bound-T User Guide [2] and Reference Manual [1] and the target-specific Application Notes by describing how Bound-T reads and uses symbolic debugging information represented in the *CDB* format, as emitted by the Small Device C Compiler, SDCC [4]. A CDB file contains only symbolic debugging information, no program code or data.

This Technical Note describes Bound-T CDB support in a generic way, without considering a particular target processor. When a version of Bound-T for a target processor supports CDB, the Application Note for that target processor may give additional target-specific details, for example on the mapping of numerical addresses to particular address spaces in the processor.

1.2 Overview

The reader is assumed to be familiar with the general principles and usage of Bound-T, as described in the Bound-T Reference Manual and User Guide [1, 2]. The user guide also contains a glossary of terms, some of which may be used in this Technical Note. You may also may find it useful to first read the Bound-T Application Note for your target processor.

The remainder of this document is structured as follows:

- Chapter 2 describes the main features of the CDB format and how they relate to the functions of Bound-T. The chapter also gives an overview of how Bound-T reads and uses an CDB file.
- Chapter 3 explains the warning messages that Bound-T may emit if it finds some problems or unsupported features in an CDB file.
- Chapter 4 explains the possible error messages similarly.

1.3 References

[1] Bound-T Reference Manual.
Tidorum Ltd, Doc. ref. TR-RM-001.
http://www.bound-t.com/ref-manual.pdf.

[2] Bound-T User Guide.
Tidorum Ltd, Doc. ref. TR-UG-001.
http://www.bound-t.com/user-guide.pdf.

[3] CDB File Format.
Lenny Story, SDCC Development Team, 2003-03-21.

[4] Small Device C Compiler (SDCC). *http://sdcc.sourceforge.net/*.

[5] ASxxxx Assemblers and ASLINK Relocating Linker.
Alan R. Baldwin, Kent State University, Version 2.0, August 1998.

1.4 Typographic conventions

We use the following fonts and styles to show the role of pieces of the text:

-option	A command-line option for Bound-T or other tools.
symbol	A mathematical symbol or variable.
`text`	Text quoted from a text / source file or command.
Record Field	The name of an CDB record (type), or the name of a field in an CDB record.

1.5 Abbreviations and acronyms

See also reference [2] for abbreviations specific to Bound-T and reference [3] for CDB terms.

CDB	C DeBug (?) [3]
Hex	Hexadecimal
SDCC	Small Device C Compiler
WCET	Worst-Case Execution Time

2 THE CDB FORMAT

2.1 CDB features

CDB

A CDB file provides symbolic names (identifiers) and some semantic type information for the subprograms and variables that form a compiled and linked (executable) program and reside at known addresses or memory locations within that program. A CDB file does not contain any machine code or initial data values to define the initial memory image for the program. In the SDCC compilation system the machine code and constant data are placed in a separate file, usually in Intel® Hex form.

In the SDCC compilation system CDB files are generated by the linker [5] using similar but relocatable files from the compiler [4] and assembler [5].

CDB records and record types

A CDB file is a text file with one line per record. There are several kinds of records with some similarities in form. Speaking broadly there are two classes of records: symbol records and linkage records. The order of records is generally not significant. When records refer to shared information they do so by name, for example by type-name.

The symbol records can:

- Name the module that defines these subprograms and variables.
- Define a function (a subprogram).
- Define a symbol (for a subprogram, a variable, or some other things).
- Define a structured type.
- Report memory layout information from the linker.

The linkage records can:

- Define the machine address of a symbol.
- Connect source-code lines to code addresses.

Bound-T uses such symbolic information to translate symbolic inputs such as subprogram names, variable names, and source-line numbers into machine-level addresses and conversely to translate machine-level analysis results into symbolic outputs.

Overall CDB structure

The order of records in a CDB file is generally not significant. However, Bound-T assumes that all references to shared information within a given record are satisfied (defined) in earlier records. Thus, all linker memory-address records (L:L and L:G records) must follow the corresponding symbol records (S, T, and F records).

2.2 How Bound-T reads a CDB file

Reading and loading a CDB file

Bound-T typically reads a CDB file as a text file in one pass from the first line to the last line. The file is expected to have the standard line terminators for text files on the current host system.

Bound-T parses each line in the file, detects its type, checks for some errors, and enters the information in its own symbol table structure. Some parts of this process depend on the target processor for which the target program is compiled.

2.3 Target-specific parts of CDB processing

The process described above for reading and loading an CDB file invokes some actions specific to the target processor for which the target program is compiled and linked. The actions that can be defined in a target-specific way are the following:

- Mapping a code address string to a program-memory address in the target processor.
- Mapping a data address string (or register list, *etc*.) to a storage cell in (Bound-T's model of) the target processor.

These target-specific steps in CDB processing should be explained in the Bound-T Application Note for the relevant target processor, as should be the warning or error messages that may issue from these steps.

3 CDB WARNING MESSAGES

The following table lists the warning messages that Bound-T may emit to highlight some problems or unsupported features in a CDB file. The messages are listed in alphabetical order. Variable fields in the message are indicated by *italic* text and ignored in the alphabetic order. The Bound-T Reference Manual [1] explains the general form of warning messages. The Bound-T Application Note for the relevant target processor may describe additional warning messages relating to the target-specific steps in CDB processing.

The probable reason for any of these warnings is either a damaged CDB file, or a file that uses a version of CDB that Bound-T does not support. To correct the problem you should obtain an undamaged program file, in a supported version of CDB, or in some other format that Bound-T supports for your target processor.

Table 1: Warning messages for CDB files

Warning Message		*Meaning*
CDB Linker symbol in *space*: *scope*: *name*	*Reasons*	A Linker record in the CDB file locates a variable in a memory space that Bound-T does not currently model.
	Action	Take note that the variable called *name*, in the given *scope*, is not available to the analysis nor in assertions.
CDB symbol "*name*" (*scope*) assumed to be unsigned octet in External RAM	*Reasons*	The CDB file contains a Linker record for the symbol with this *name*, in this *scope*, but does not contain a (preceding) compiler-generated record giving the type of the named entity. Bound-T assumes that the name represents an unsigned 8-bit variable (char) in the external data memory at the address given in the Linker record. This warning is usually optional and appears only if you run Bound-T with a specific command-line option (typically *-warn cdb_def*).
	Action	If you use this variable (*name* and *scope*) in an assertion, check that the assumption on the variable's type and location holds. If the variable has some other type (especially if it has another size) an assertion using this variable may have an unintended effect on the analysis. Use the indicated command-line option (typically *-warn no_cdb_def*) to disable this warning.

4 CDB ERROR MESSAGES

The following table lists the error messages that Bound-T may emit to highlight severe problems or unsupported features in a CDB file. The messages are listed in alphabetical order. Variable fields in the message are indicated by *italic* text and ignored in the alphabetical order. The Bound-T Reference Manual [1] explains the general form of error messages. The Bound-T Application Note for the relevant target processor may describe additional error messages relating to the target-specific steps in CDB processing.

The probable reason for any of these errors is either that the CDB file is damaged or that the file uses a version of CDB that Bound-T does not support. To correct the problem you should obtain an undamaged program file, in a supported version of CDB, or in some other format that Bound-T supports for your target processor. If the CDB file is correct, please report the problem to Tidorum.

Table 2: Error messages for CDB files

Error Message		*Meaning*
CDB: *message*: *rest of line*	*Problem*	While parsing a line in a CDB file [3] Bound-T found the problem described in the *message*, a problem with the syntax or contents of the line. The unparsed part of the line is shown as *rest of line*.
		See Table 3 below for the possible *messages* and their meaning.
	Reasons	The CDB file is damaged, or uses a form of CDB that Bound-T does not support.
	Solution	If the CDB file is correct, please report the problem to Tidorum.
CDB file not found: *name*	*Problem*	Bound-T tried to open and read a CDB file with this *name* but could not find a file with this *name*.
	Reasons	Perhaps the file *name* is mistyped (if given as input); perhaps the file is in a directory that is not accessible.
	Solution	Correct the file *name* if in error. Set directory access rights to allow access to the file.
CDB file not readable: *name*	*Problem*	Bound-T tried to read the CDB file with the given *name*, and a file of this *name* seems to exit, but Bound-T failed to read it.
	Reasons	The file may be read-protected (insufficient access rights) or the *name* may refer to a directory or some special file that cannot be read.
	Solution	Correct the file access permissions or correct the file-name.
CDB line ends short: *line*	*Problem*	While parsing a *line* in a CDB file Bound-T found that the line ends unexpectedly, omitting some text that the CDB syntax requires for this line.
	Reasons	The CDB file is damaged, or uses a form of CDB that Bound-T does not support.
	Solution	If the CDB file is correct, please report the problem to Tidorum.

Error Message		*Meaning*
CDB line not understood: *line*	*Problem*	While parsing a *line* in a CDB file Bound-T found some unspecified problem with the syntax or contents of the *line*.
	Reasons	The CDB file is damaged, or uses a form of CDB that Bound-T does not support.
	Solution	If the CDB file is correct, please report the problem to Tidorum.

Table 3 below lists and explains the *message* fields that can appear in error messages of the form "CDB: *message*: *rest of line*".

Table 3: CDB parsing errors

The "message" field	*Meaning*
AddressSpace conflicts with OnStack	In this Symbol or Function record the values of the <Address Space> field and the <On Stack> field are not compatible. The <Address Space> values 'A' and 'B', and no others, mean that the variable is in a stack. (Thus, the <On Stack> field is really redundant in these records.)
Function has non-code AddressSpace	This Function record defines a function (a subprogram) but the <Address Space> field has a value other than 'C', 'D', or 'Z', which are the only address spaces valid for functions.
Expected '*c*'	The character *c* was expected at this point in the CDB line, but some other character was found. This message typically means that some required delimiter is missing in the line.
Expected end of line	The current CDB line was expected to end at this point, but some text remains in the line.
Unknown AddressSpace code '*a*'	The <Address Space> field in this Symbol or Function record contains the character *a* which Bound-T does not recognise as a valid address space code.
Unknown IsInterrupt code	The <Is Interrupt > field (also known as the <Interrupt> field) of this Function record contains a character which is not one of the two valid choices '0' and '1'.
Unknown Linker record code '*t*'	The record-type field (the character after the initial "L:") of this Linker record contains the character *t* which which Bound-T does not recognise as a valid Linker-record type.
Unknown OnStack code	The <On Stack> field of this Symbol or Function record contains a character which is not one of the two valid choices '0' and '1'.
Unknown record-type code '*t*'	The record-type field (the first character) of this line in the CDB line contains the character *t* which which Bound-T does not recognise as a valid CDB record-type code.
Unknown scope code '*s*'	The scope field in this Symbol or Function record starts with the character *s* with is not one of the defined scope codes 'G', 'F', 'L'.
Unknown signedness code	The <Sign> field in this Type Chain record contains some character other than the valid signedness codes 'U' and 'S'.
Unknown type "*t*"	The data-type designation field (<DCLType> in [3]) contains the value *t* which does not designate a type that Bound-T knows about.

Tidorum Ltd

Tiirasaarentie 32
FI-00200 Helsinki
Finland

www.tidorum.fi
info@tidorum.fi
Tel. +358 (0) 40 563 9186
Fax +358 (0) 42 563 9186
VAT FI 18688130