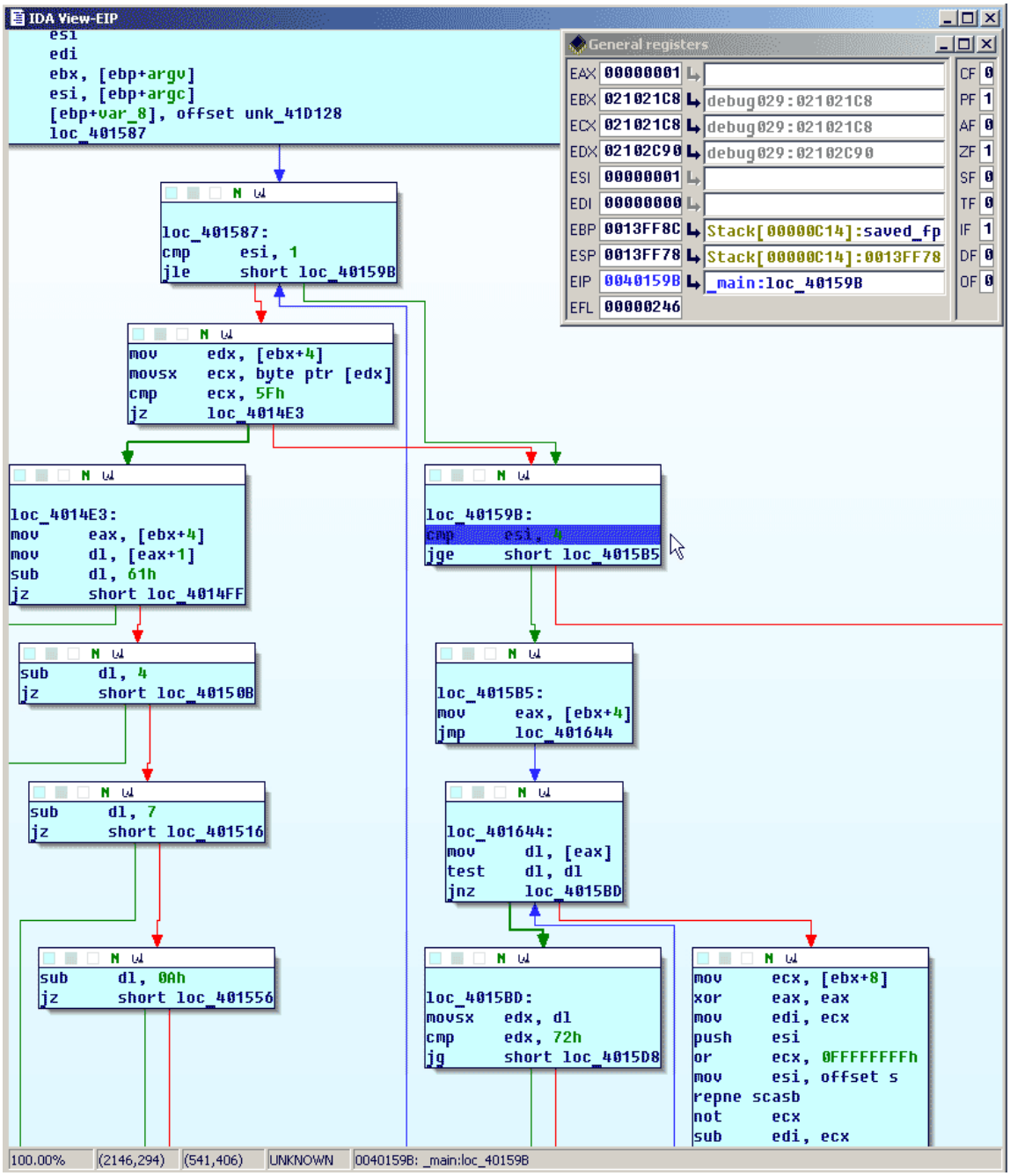
IDA Pro 5.0 (March 2006)

Major features

- Introduction of a graph based used interface. The text interface remains instantly available.



Processor Specific Enhancements

- ARM: improved distinction of code and data: conditional instructions do not start a new function.
- ARM: IDA knows that a function call destroys R0.
- ARM: IDA knows that only GNU AS reverts halves of double data items; for other assemblers the double number format conforms the standard (IEEE).
- ARM: IDA tries to find out the base register of the stack variables by looking for 'mov rN, SP' instructions.
- ARM: MOV R12, SP is recognized as the beginning of a code sequence.
- ARM: new target assembler: ARM/Thumb Macro Assembler.
- ARM: slightly better jump table recognition.
- **JAVA: complete rewrite of the Java module to support the new JDK 1.5 (or Java5.0)**
- PC: added support for the newly documented 'cmpxchg16b' instruction.
- PC: improved function analysis.
- PC: better test of instruction sanity.
- PC: ins instruction was always displayed in the long form.
- PC: more careful approach to jump table xref construction.
- PC: previously undocumented form of the 'test' instruction is recognized (group 3modrm /1)
- PC: newer versions of SEH_ prolog/epilog functions are recognized
- 6812: the HCS12 config file has been updated
- 78k0: has been replaced by a rewritten module
- 78k0s: has been replaced by a rewritten module

File Formats

- ELF: added support for SPARC unaligned relocation types.
- ELF: relocations in .gnu.conflict section are ignored since this section is not loaded by default.
- COFF: MC68K: support for R_PCR24 relocation type has been added (used in PalmOS).
- DBG: ida does not create functions for data names.
- more PalmPilot system trap codes are added.
- if the input file is corrupted, IDA displays an error message without exiting to the OS.

Kernel Enhancements

- DDK2003 type library files have been updated; wnet/windows.h types have been added.
- Flow charts of processors with delayed jump slots are generated correctly (this feature requires support from the processor module).
- a regular function is created instead of a function tail if it makes sense.

- analysis: the rule which creates functions because of a dref has been improved.
- better use of fixup information during the final pass of the analysis.
- FLAIR: CodeWarrior library files for 6812 are supported (since the file format is undocumented, there might be problems).
- IDA does not automatically assign a type to local names because it rarely makes sense
- recognition of function pointer tables has been improved.
- turning off the solid border lines turns off SUBROUTINE lines too.
- a full path is accepted in ida.cfg:GRAPH_VISUALIZER.
- minor improvement of switch table construction (if a jump table crossed through segment boundaries, IDA would fail to create it)
- **signature files have been updated or added: Borland Developer Studio 6, Microsoft Visual C runtime version 8 (.net) 32-bit and 64-bit libraries, Microsoft MFC 64-bit, Microsoft Active Template Library 64-bit.**
- the MD5 of the input file is saved in the database.

IDC & SDK

- IDC: renimp.idc: is a new script that renames import table entries.
- IDC: the SetType() function can be used to delete the existing type assigned to an address.
- IDC: SetSegmentAttr() accepts SEGATTR_BITNESS attribute and changes the segment bitness without reanalyzing it.
- SDK: calc_bare_name() has been improved to handle __imp_ and c++ mangled names.
- SDK: guess_func_type() takes into account the number of purged bytes from the stack: if the tail parameters were not used by the function and therefore were not created by IDA, we still create dummy arguments for the in the function type.
- SDK, IDC: del_segm() accepts a combination of bits as the second parameter.
- SDK: added a flag to flow_chart_t to avoid computing external blocks.
- SDK: added processor_t::gen_asm_or_lst to customize asm or lst file generation.
- SDK: added processor_t::is_insn_table_jump to determine if an instruction is really a table jump or call.
- SDK: added SDL_HIDETYPE bit for segments – it is used to hide the segment type from the disassembly listing.
- SDK: added ui_create_tform and other callbacks to manipulate MDI child windows from plugin.
- SDK: analyze_area() function can be applied to debugger segments as well; before it was skipping them.
- **SDK: an API to work with graph viewer is added. See the sample plugin ugraph**
- SDK: areacb_t::for_all_areas() function to enumerate all areas in the specified range.
- SDK: autoIsOk() would return false for old database when called from ph.oldfile
- SDK: callback out_src_file_lnnum to generate source file name and line number directives.

- SDK: if inf.lowoff == BADADDR, no operand will be considered as 'void' operand.
- SDK: if Namechars[] is empty, all characters are enabled in names.
- SDK: if public or weak keywords are defined as empty strings, then IDA does not display the corresponding directives.
- SDK: introduced new event processor_t::auto_empty_finally to handle the end of autoanalysis for efficiently.
- SDK: new function entab() to replace spaces by tabulations.
- SDK: new function qmake_full_path()
- SDK: ph.get_autocmt notification to generate dynamic predefined comments for instruction.
- SDK: new function get_compiler_name()
- SDK: added CH_MULTI_EDIT bit for the list choosers.
- SDK: added read_user_config_file() function.
- SDK: loader_finished event has been added.
- SDK: **4 new processor modules and their source code have been donated by a kind IDA user: Toshiba TLCS-900, Rockwell C39, NSC CR16, Panasonic MN10200**

User Interface

- GUI: the analysis indicator is refreshed at most 10 times per second.
- GUI: the keypad 5 scrolls the window to center the keyboard cursor.
- GUI: the Ctrl-F/F3 hotkeys search in the database notepad.
- the input fields of most dialog boxes are remembered in the registry and database; database settings have priority over registry settings; TEXT_SEARCH_CASE_SENSITIVE and BIN_SEARCH_CASE_SENSITIVE are removed from the configuration files; added RESTORE_UI_VARS and USE_INIFILE user interface config parameters.
- it is possible to delete marked positions from the 'jump to marked position' dialog box.
- UI: 'search for all occurrences' flag works in the selected area if there is any.
- UI: 'set type' command works with a location in the middle of a function if the location already has a type; otherwise it is applied to the whole function.
- UI: the text version asks the permission to destroy the existing items if they prevent the creation of another item specified by the user; the config file parameter is AUTO_UNDEFINE
- wingraph32 related commands are now available for all platforms (Linux, Windows)

Debugger

- debugger colors do not override item colors anymore.
- debugger: start the application in its own directory by default if not instant debugging.
- **debugger: debugging is supported in graph mode.**

Bug Fixes

- the "function calls" window was not saved/restored in the desktop configuration; its name in the tab control was wrong (had function names)
- the "incompatible main desktop config" message has been removed; such desktops are now silently ignored.
- the 64-bit debugger did not understand register names in idc expressions
- a corrupted database with -1 as the assembler type could crash IDA
- if turned off the analysis indicator in the options dialog box would read 'idle' instead of being empty.
- analysis could loop infinitely on some files.
- clicking Close in the taskbar at the the startup screen or welcome dialog could crash IDA
- closing the 'function calls' window would not delete the corresponding menu item in Windows men.
- corrupted DBG files could crash IDA.
- debugger: terminating multithreaded applications required several attempts.
- HTML files generated from an automated IDC script always had a black background.
- IDA could display a message asking the permission to delete debug segments and later fail because the answer came too late.
- if IDA had been installed in a C:\Program Files subdirectory, launching wingraph32 could lead to the execution of c:\program.exe (if present)
- in 64-bit mode IDA could display an instruction with a floating point register fp(8) or higher
- in MS DOS COM files it was impossible to use offsets based on the beginning of the first segment
- it was impossible to run an IDC script using the script toolbar if there was no open database
- JAVA: it was impossible to use IDC in the graphical version.
- memory hex dump files without the address column were loaded incorrectly.
- pfn pointer could become stale during function chunk enumeration leading to wrong flow charts.
- REX prefix should not modify AL register in most AMD64 instructions.
- the 'print flags' command was not correctly displaying national characters in the comments.
- the analysis could infinitely loop on garbage bytes looking as legitimate code.
- the analysis pointer in the navigation band stayed visible even after end of the analysis (until the first refresh).
- IDA could crash if the input file could not be opened (blocked by an antivirus, for example)
- the 'rename register' command would an cause 'internal error' if the old register name was empty.
- the help page about maximal address space was missing from the help file.
- A problem in the database naming logic after an unclosed debugging session was fixed.

- the 64-bit text version was displaying zeroes in the autoanalysis indicator (in fact, the upper part of the address). Switched to the low part since it gives more information

Contact Information

© 2006 DataRescue SA/NV

40 Blvd Piercot

4000 Liège

Belgium

t - +32-4-3446510

f - +32-4-3446514

info@datarescue.com