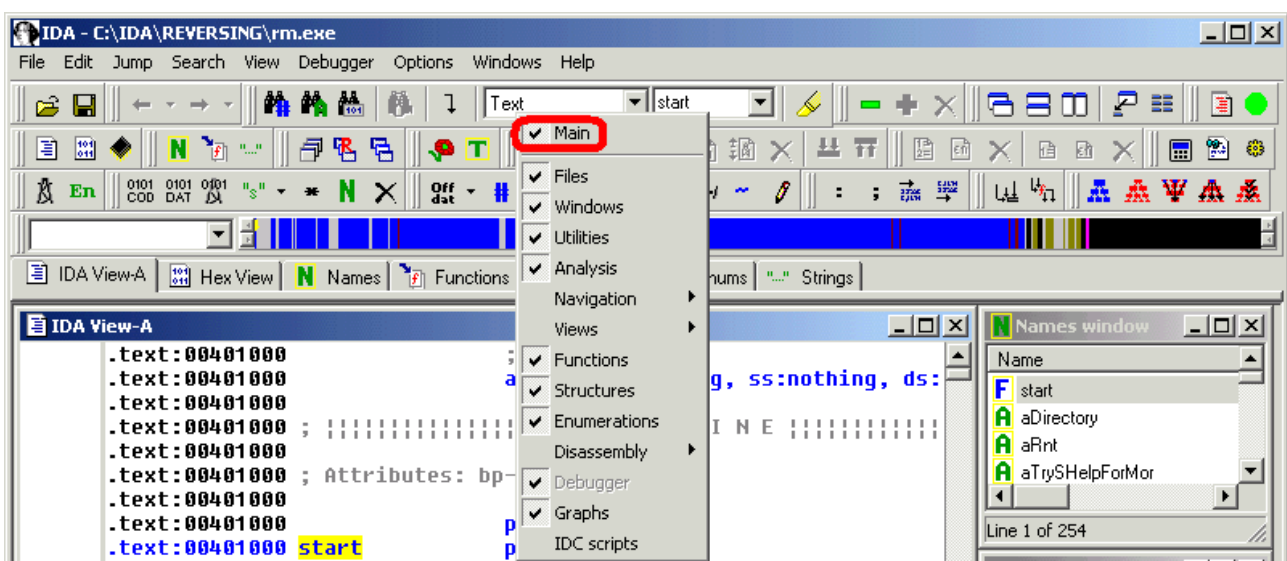


Customizing IDA Pro. © DataRescue 2005

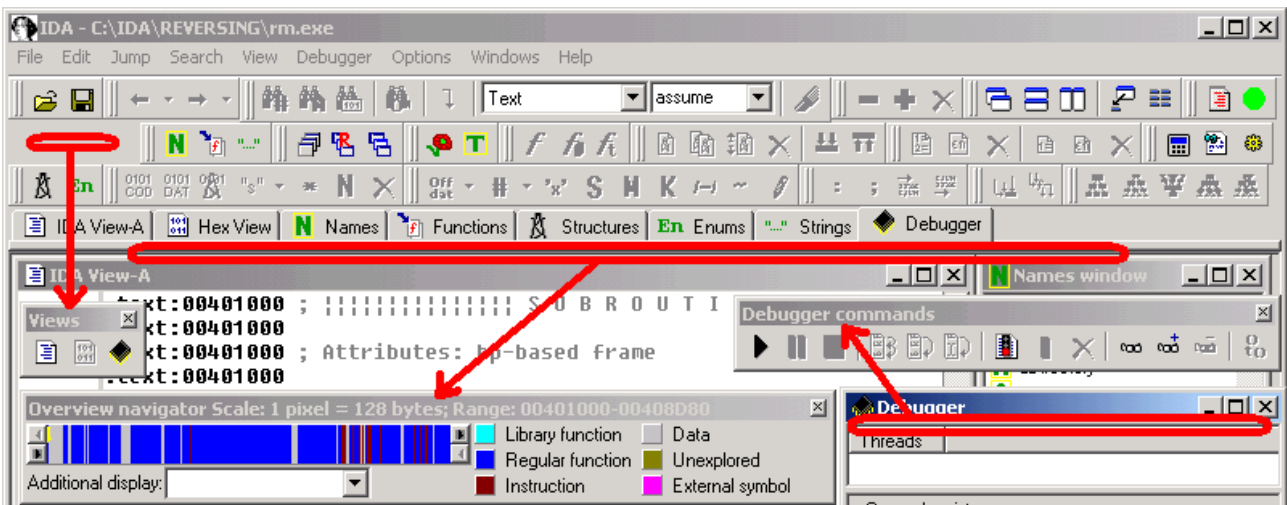
Since version 4.30, IDA Pro supports a new MDI tabbed interface. Here are a few tips to customize it.

Toolbars.

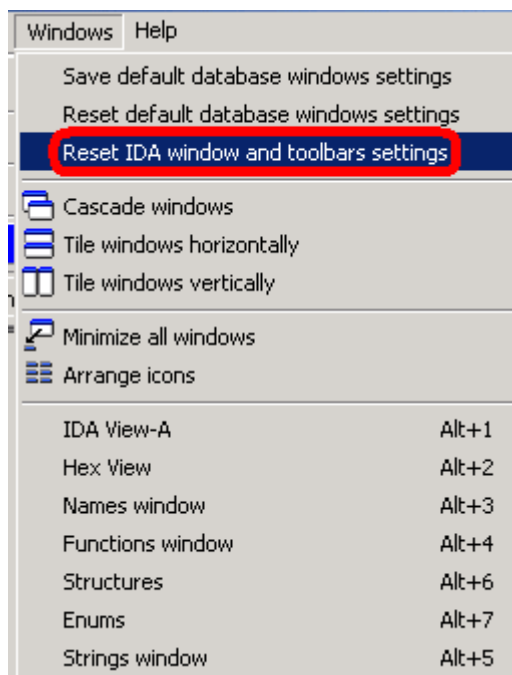
Open the popup menu by clicking the right mouse button on the panel containing all the toolbars: here you may hide all toolbars at once, by clicking on the *Main* command, or selectively hide rarely used toolbars.



By default, most toolbars are docked to a reserved area just below the main menu bar. But you can also undock toolbars, and put them anywhere on your screen. The *Debugger* toolbar is a bit special: by default, it is docked in the *Debugger* window. If you wish, you may undock from the debugger window and dock it to the panel containing all others toolbars, or set it as floating on the screen.

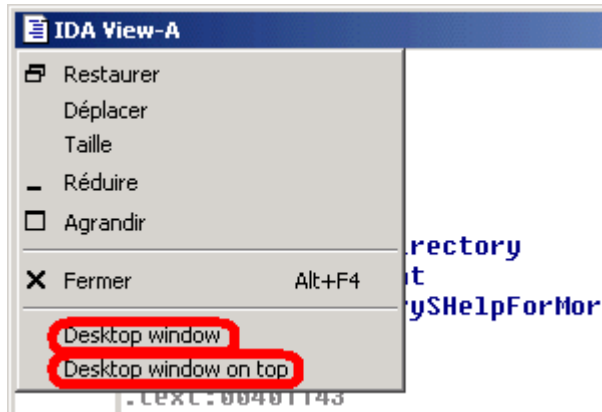


The configuration of IDA toolbars is automatically saved in the registry. If for some reason, you wish to restore their original configuration, simply click on the the *Reset IDA window and toolbars settings* command.



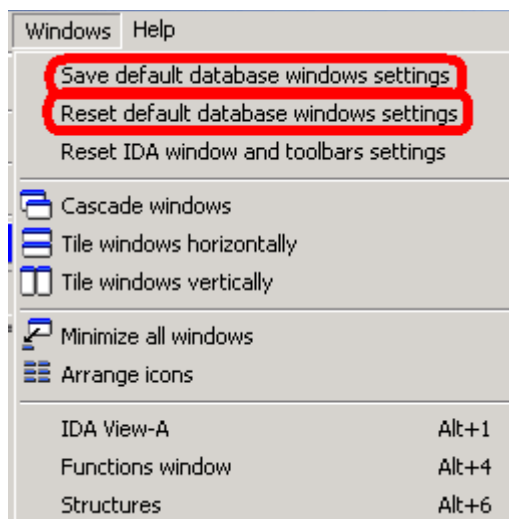
Windows.

IDA creates new windows as a MDI child windows. If you prefer, you may also put those windows on the desktop (outside of IDA main window), or on top of your desktop. For this, open the window's system menu, and select the requested option.

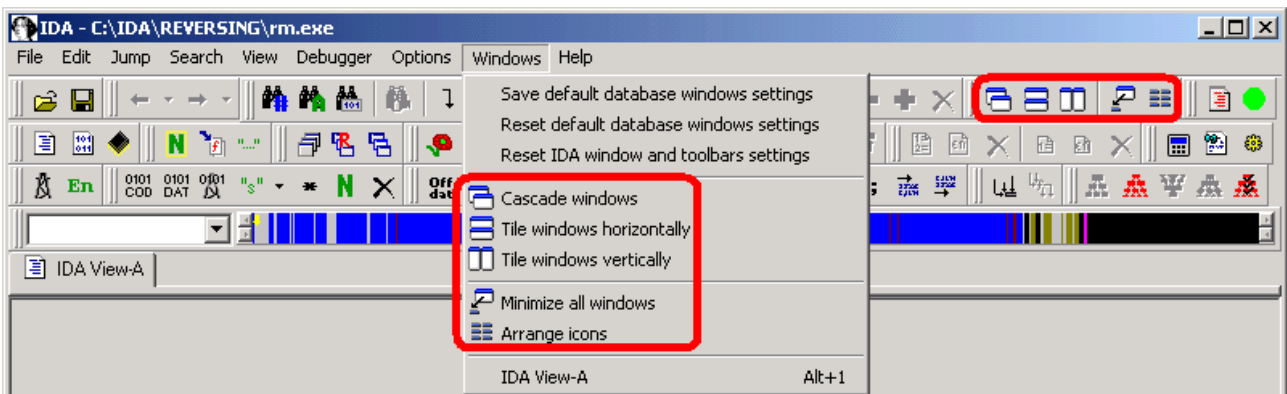


The state (MDI / desktop / on top), the position and the dimension of windows are saved in the database itself.

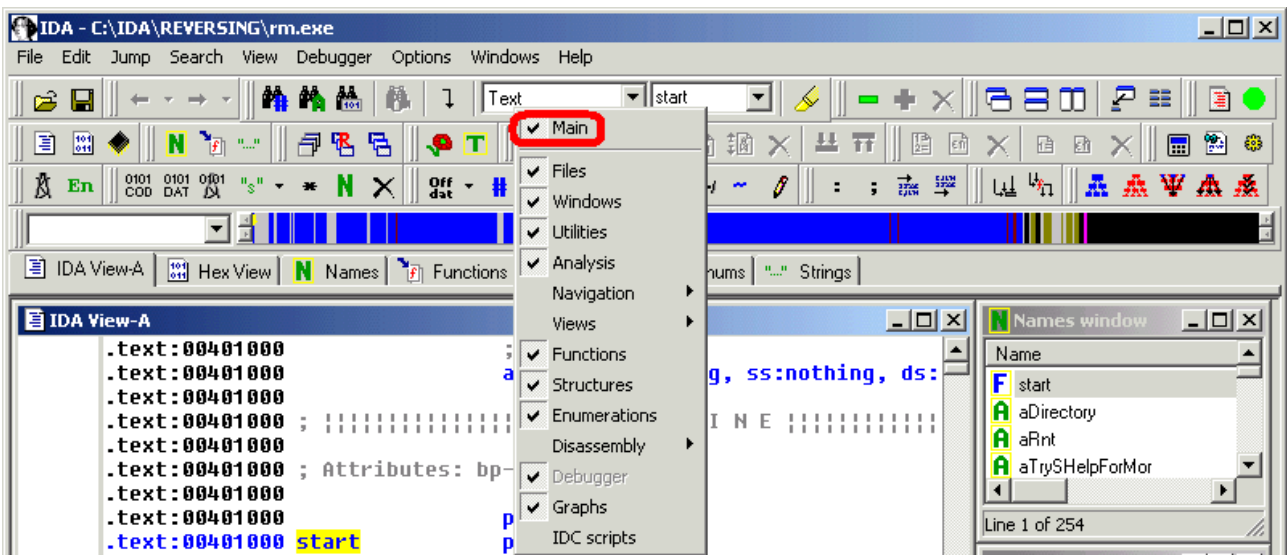
When the user creates a new database, IDA automatically opens many windows. You may define the default set of Windows that IDA will open once it creates a new database. For this, simply adjust windows configuration to your liking in an existing database, and use the Save default database windows settings command. From now on, IDA will use this configuration for all future databases it creates. If you want to remove this default configuration, simply click on the *Reset default database windows settings*.



The *Windows* toolbar or the associated commands in the *Windows* menu allow you to define the appearance of your MDI windows.



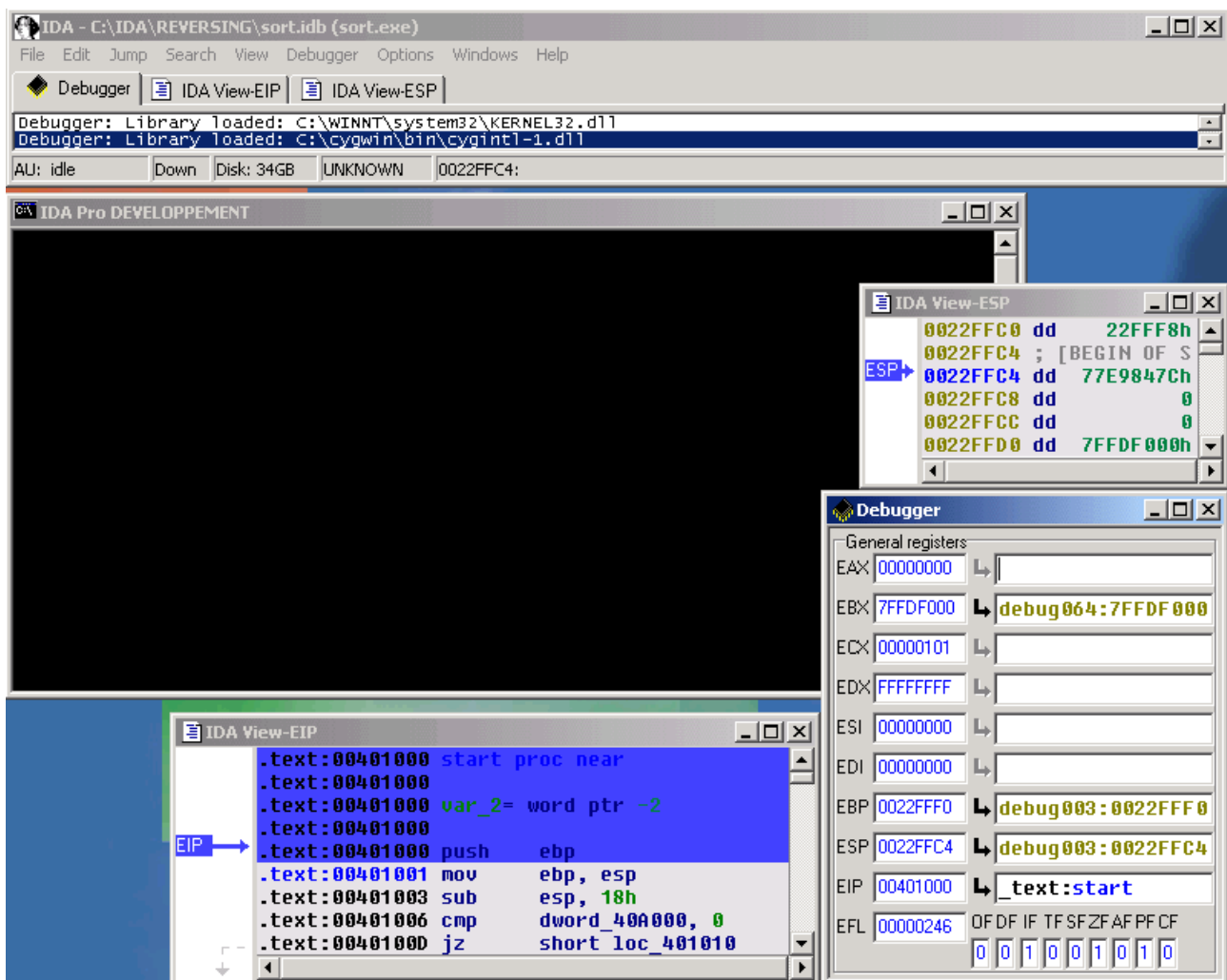
To switch from one window to another, IDA associates each window a small tab. Right clicking on this tab opens a popup menu, allowing you to close the window, or set it up as MDI / desktop / on top (as presented previously).



Let's now apply those toolbars and windows related tips to create some interesting configurations in well defined cases.

Debugger.

When we use a debugger, it usually needs as much screen as possible for the debugged application itself. How do we configure IDA in manner which that is as little intrusive as it is possible on the screen? First, we can modify some *Disassembly Options*, such as removing *Instructions indentation*. In the debugger window itself, we can hide the threads list and the debugger toolbar, by unselecting the corresponding items in the associated popup menu. We could also hide all others toolbars. Just before starting the debugged process, we close all windows, except the debugger window. We start the debugging session, set the *IDA View-EIP* and *IDA View-ESP* windows as desktop windows, and resize them to our taste. Finally, we can reorganize IDA main window to only display the menu, the tabs and the messages panel.



Keyboard.

The IDA GUI configuration file can be edited to assign hotkeys to almost all available commands. For example, the following section represents the default hotkey definitions for debugging commands.

```
//
//      Debugger manipulation commands
//

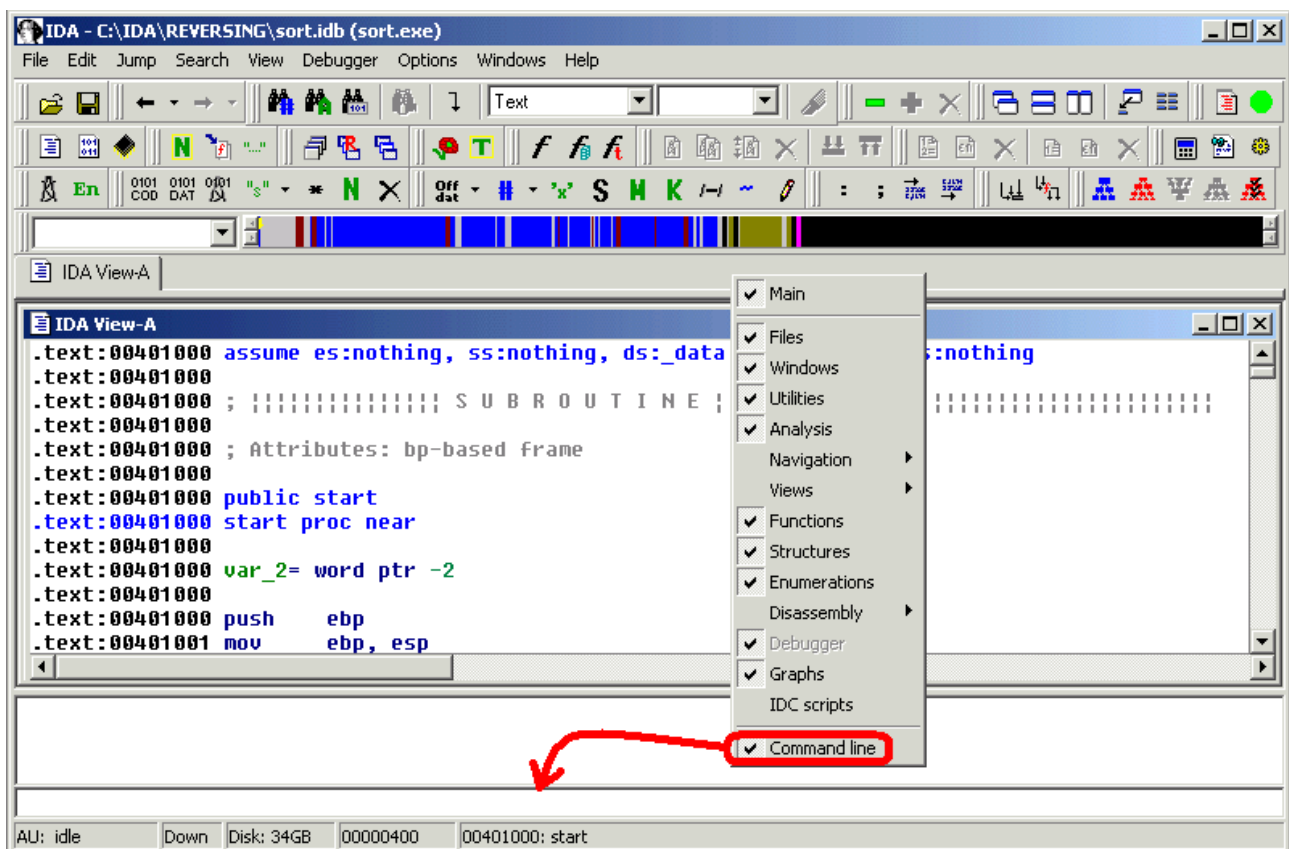
"Debugger"           = "Ctrl-Alt-C" // open debugger window
"ProcessStart"       = "F9"         // start a new process in the debugger
"ProcessPause"       = 0           // pause the debugged process
"ProcessExit"        = "Ctrl-F2"    // terminate the debugged process
"ThreadTraceInto"    = "F7"         // trace into the current instruction
"ThreadStepOver"     = "F8"         // step over the current instruction
"ThreadRunToCursor" = "F4"         // execute instructions until cursor is reached
"BreakpointAdd"      = 0           // add a breakpoint
"BreakpointDel"      = 0           // del a breakpoint
"BreakpointToggle"   = "F2"        // toggle a breakpoint
"BreakpointEdit"     = 0           // edit breakpoint settings
"BreakpointEnable"   = 0
"BreakpointDisable" = 0
"Breakpoints"        = "Ctrl-Alt-B" // open breakpoints window
"WatchList"          = "Ctrl-Alt-W" // open the watch list
"AddWatch"           = 0           // add watch
"DelWatch"           = "Del"       // del watch
"StackTrace"         = "Ctrl-Alt-S" // open stack trace window
```

Command line.

The *Main Window* offers a command line that allows you to invoke IDC function or execute calculations. To make it available we need to edit the IDAGUI.CFG file. We simply change the following line:

```
DISPLAY_COMMAND_LINE = YES // Display the expressions/IDC command line
```

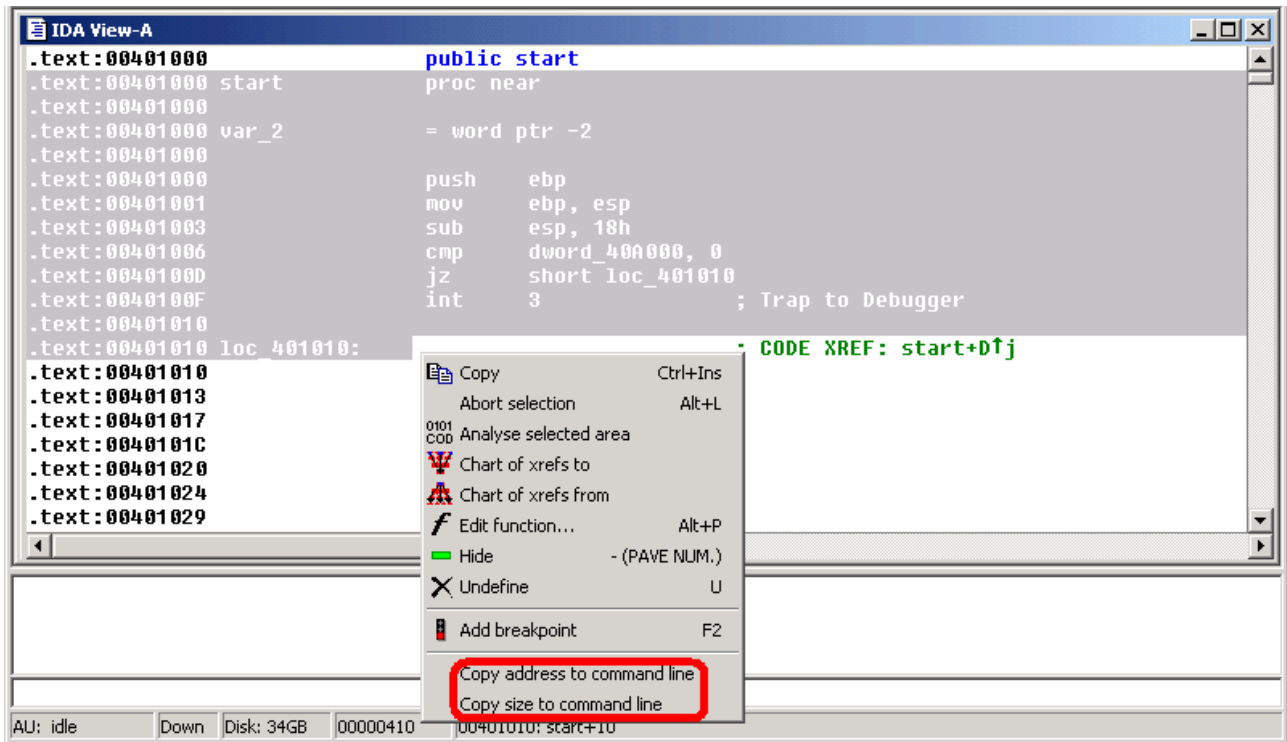
This will add a *Command line* item to the toolbars popup menu. We can now display the command line by clicking on this item.



The user can use the command line to evaluate expressions. Expressions can be classical mathematical expressions but may also include calls to IDC functions. If you put a ';' char at the end of the expression, IDA will not print the return value in the messages panel. To make the edition of complex expressions easier, we can use new commands available in the disassembly view popup menu.

The *Copy address to command line* command will put the current address in the command line. If a range is selected, the *Copy size to command line* command will put the size of the current selected range in the command line.

In both cases, IDA will try to intelligently insert spaces or commas to obtain a valid expression. Up/down arrow keys can be used to browse among previous entered expressions. The user interested in IDC scripting can also press the TAB key to auto-complete IDC function names, or press the F1 key to obtain help on a particular IDC function.



As you can see, the IDA environment is fully configurable.

This tutorial is © DataRescue SA/NV 2005

Revision 1.1

[DataRescue SA/NV](#)

40 Bld Piercot

4000 Liège, Belgium

T: +32-4-3446510 F: +32-4-3446514