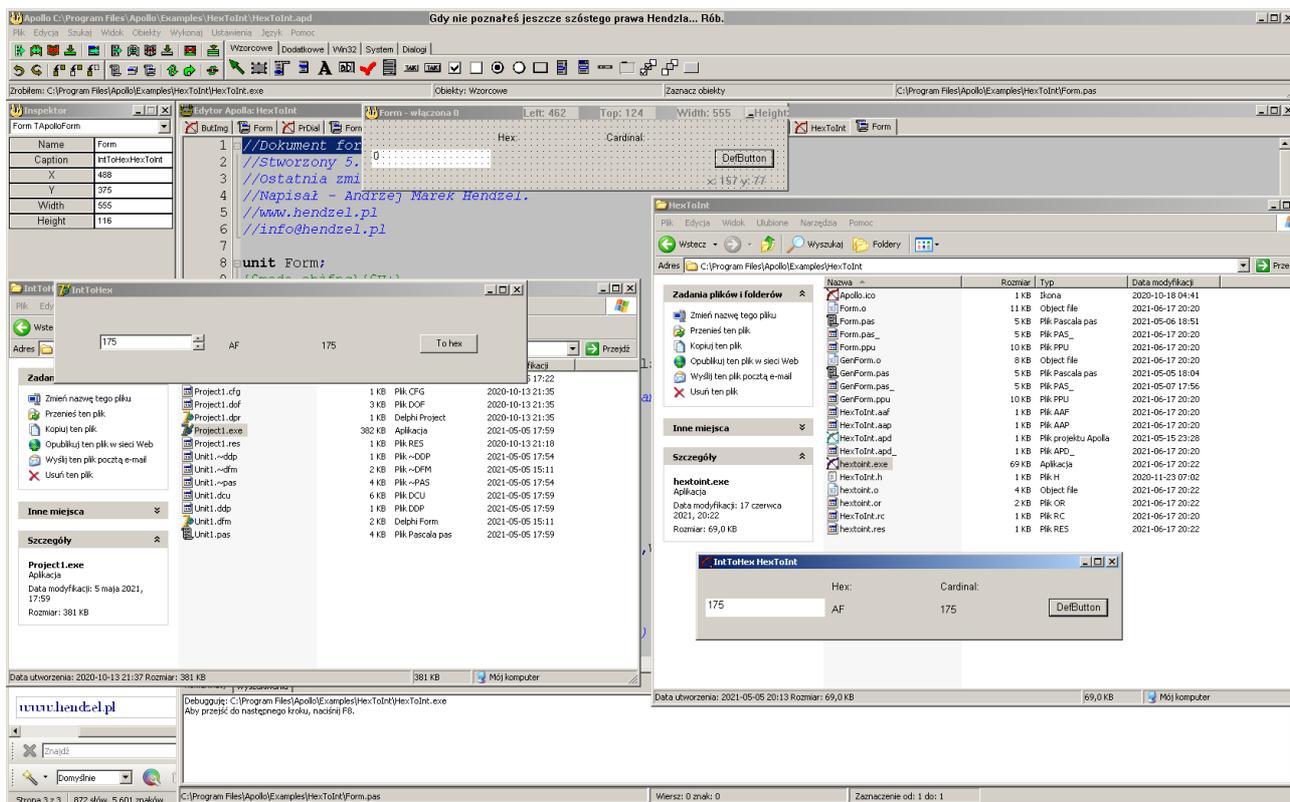


Apollo Was Born in Apollo

Yes, it is possible. Apollo had originally been an ordinary text-editor drawn from the program Dev-Pas. It compiled itself from the file BAT (*.bat). After the addition of a start-up module and a simple button in Compile menu, it had begun to compile itself on its own. And thus was born a nucleus from which, step by step, with every change of the source to be opened and edited up in Apollo, a new Apollo was being constantly created. One can say that it grew like a snowball, or even better, like a living creature, expanded and constantly improved.

There are people perpetually dissatisfied, especially at the achievements of others. But supposing that instead of nitpicking and complaining at what isn't there, they would notice what is there. Such people are like those boarding a plane and grumbling at being forbidden to open a window, when they could do so in a car. They could do so in a carriage, too. And what a drive it is on a horse-drawn cart with wind-blown hair. Yet hardly at such speeds and at such altitudes as those of today's ordinary passenger planes with their hermetic cabins preventing you from suffocating in the thin air.



the class of the application `WinRegister()`. Thanks to this file the programmer knows what the application is really doing. Is it still little?

There are those for whom little is too little? Let us add something more for them, then. In Apollo there is, for example, a function `HexToInt()`. It has a little misleading name, as in fact it changes hexadecimal values into natural variables rather than generally whole ones, but it does it adroitly. In addition, the values written into the box to be changed first into a hexadecimal variable may contain non-digit inclusions, and yet Apollo will convert them correctly into a number and write them down in hex. Whatever one chooses. And the same will be done by the function `HexToInt()`. For example, from the format `$00AF`, or `#00AF`, or `00AF`, or ultimately `AF` the change will be made into the correct value of the natural number 175. It is an exceptionally flexible method. It does not return errors wantonly as do Delphian or Lazarusian functions.

Apollo has a lot of such functions. Not all of them are fully tried out, and so may return some errors. However, Apollo is currently a program in Hiper Beta version... which can be followed on the website www.hendzel.pl and in Apollo itself. Whereas the hexian applicationlet itself lies in the directory (though most of the unsatisfied prefer the name "folder") `Examples/HexToInt`. Which can be seen in the picture. In other words, it is delivered with Apollo as an example in the form of the project's source code along with many other examples

Apollo does not by default save the project nor any file in the course of compilation – which means, prior to it. The compiler downloads the values for compilation always from the source file, so it is easy to guess that the source must be saved before compilation, in order to make apparent the result of the source change. Why doesn't it do it? So that the programmer develops the habit of saving his work as it progresses, so as not to have it lost. The automaton makes you lazy. And it sometimes happens that you tap in the code while at work, in the understanding that the compilation button will settle everything and save the source file. And then bang – the electricity breaks down or the UPS or laptop battery goes. He who never lost the results of his work does not know what it's like. That's why it's better to have a state of reflex – if something important is tapped in, I save the file. It's like using the indicator. "Eh, why bother with it when the roadway is empty." – and so it goes. Up till the time when you find yourself in heavy traffic, to end up in another (though perhaps a bit heavier) bang. Because of getting accustomed to the idea that the environment itself will save for you the source file. Does anyone think it takes much work to paste along the saving prior to compilation? It takes twenty seconds to do it. But a bad habit is a bad habit, and we want to have good habits.

There are many such places in Apollo. And the environment benefits from this in being useful and speedy. It has a small palette of components, but here one has to wait. There's no rush. Most important are the mechanisms of work, code analysis and compilation quality. And there are more urgent needs than building away palettes. On palettes currently half of Poland is riding, for it looks as if we are capable of doing nothing except

hammering about palettes and then trundling someone else's commodities on them. And doing it in foreign parts, to boot. Which is not only not like America, but first and foremost not like Poland.

God bless.

Andrzej Marek Hendzel