M$ Announces DRM-Oriented Programming Language

By *srlinuxx*
Created *21/06/2005 - 2:50pm*
Submitted by srlinuxx on Tuesday 21st of June 2005 02:50:09 PM Filed under Humor [1]

No programming language is complete without its own set of paradigm-enriched buzzwords, and that's exactly what Microsoft has promised with its latest gift to the programming world, code-named Freedom Unencumbered (or FU for short).

Some of the innovative buzzwords to be introduced by FU include "rights-oriented programming", "freedom-unencumbered development", "need-to-know-basis code reuse", "plug-and-pay interfaces", and "license-centered software distribution."

As these phrases suggest, FU is built from the ground up around cutting-edge DRM (Digital Restrictions Management) techniques to prevent unauthorized code reuse and idea theft.

Boasted a Microsoft product manager, "Just as C# brought object-oriented programming to the masses, FU will deliver rights-oriented methods to everyone. At a fundamental level, programming is all about balancing copyrights, licenses, freedoms, restrictions, royalties, and rights. FU makes it easy to handle all of these issues."

While the programming language is still in Beta (translation: Vaporware), Microsoft has made preliminary documentation available under its standard you-must-sell-your-soul-first license agreement (patent pending). Not wanting to agree to these terms, the Humorix Vast Spy Network(tm) was still able to obtain some whitepapers using our standard digging-through-trash-barrels-behind-the-Microsoft-campus method (patent not pending).

From what we can gather, every block of FU code must include a DRM class that spells out all of the copyrights, patents, and trademarks that the owner asserts over the code. It also states exactly what the end-user can and can't do (with emphasis on the "can't" part).

The language itself is mostly a collection of keywords and syntactical sugar that handle all of the DRM aspects. For everything else that involves actual programming work, the run-time engine merely outsources it to the Windows system libraries -- but only if all of the relevant DRM restrictions are met.

Here's the obligatory example of a Hello World program in FU:

helloworld.h:

```
drm class HelloWorldEndUserLicenseAgreement {
copyright {
covers: everything;
```

```
owner: "Bob R. Schmuckley, 313 Adam Smith St.,
Redmond, Washington";
all-rights-reserved-by-default: yes;
}
eula {
allow-redistribution: no;
allow-benchmarks: no;
allow-backup-copies: no;
total-allowed-installations-per-license: 1
agreement-method: click-wrap-license;
automatically-generate-license-text: yes;
}
warranty {
provided-as-is: yes;
warranty-of-fitness-for-a-particular-purpose: none;
warranty-of-merchantability: none;
warranty-of-title: none;
user-must-waive-all-rights-to-sue: yes;
}
enforcement {
reserve-right-to-hold-unannounced-audits: yes;
violators-to-be-prosecuted-to-fullest-extent-of-law:
well-duh;
}
output {
restrictions-on-program-output: none;
// We'll be generous here
}
fees {
payment-scheme: pay-per-use;
charge: 0.10;
currency-accepted: us-dollars;
payment-methods-accepted: credit-card;
require-user-to-prepay: yes;
}
}
```

helloworld.fux:

```
include "helloworld.h"
require HelloWorldEndUserLicenseAgreement;

not-public static void Main() {
System.Console.WriteLine("Hello, World!");
}
```

Ordinarily, most Hello World programs spit out their payload and then successfuly exit. But with FU, it's a little more complicated. Here's the process followed by the run-time engine:

[Punchline](#) [2].

[Humor](#)