



# **Free Software: A Viable Model for Commercial Success**

**Robert Dewar**  
President/CEO AdaCore

**GNU Tools Cauldron**  
Prague, July 9<sup>th</sup> 2012

# Who is AdaCore?

- **Two companies**
  - Ada Core Technologies (based in New York)
  - ACT/Europe (based in Paris)
- **These companies are separate**
  - Legally and economically
  - By informal agreement world is divided in two
- **The two companies work closely together**
  - As a single unit technically, from the point of view of customers
  - Completely unified technical development and support
- **Both companies dedicated to FLOSS**
  - For us “Freely Licensed Open Source Software”
  - All products under GPL or other free licenses
- **About 30+ people each side, total is 60+**
  - About 10 FTE marketing, sales, administration
  - Remainder are technical

## What do we do?

- **Original concentration was Ada 95 compilers and development systems.**
- **Distributions available for multiple targets on both open and non-open systems.**
- **We cover native and embedded systems. In the embedded arena we support RTOS configurations (notably partnership with Wind River for VxWorks) and bare board configurations.**
- **We have continued the strong Ada emphasis (GNAT Pro 7.0.1 just released includes virtually all of Ada 2012, so far the only compiler to do so).**
- **But we have branched out more generally**

## What we do now

- **We support C and C++ as well as Ada**
  - But currently this is aimed primarily at Ada users writing mixed language programs, though we do have some C++ only customers.
- **Rich set of tools for Ada development**
  - GPS, GNAT Programming Studio, a general IDE, focused on Ada, but also supporting other languages.
  - GNATBench, an Ada development plug in for ECLIPSE
  - Many other tools for metrics, style checking, stack checking, coverage analysis, etc.
- **Safety and Security Focus**
  - Certification materials for small libraries
  - Working with customers on large certified Avionics
  - Partnership with Praxis
  - Focus on formal verification
  - Specialized coverage tools for certification use
  - Source/object traceability studies

# Our Customers

---

- **Avionics, commercial and military**
  - Boeing (787 dreamliner common core)
  - Airbus (Thales A350)
  - Embraer
  - Lockheed
- **Defence systems**
  - Raytheon
  - EADS
  - GE (C-130 modernization)
  - Boeing (tanker program)
  - BAE (Eurofighter)
- **Space systems**
  - Lockheed (Canadian Space Arm)
- **Air traffic control**
  - Eurocontrol
  - NATS (new ground based ATC system for UK)

# Our Business Model

---

- **Sell software**
  - Includes licenses
  - Updates
  - Support
- **High end product**
  - Intensive support
  - Complex large scale development systems
  - Minimum pricing is in the 20K Euro range/year
- **AdaCore founded in 1994**
  - European company a couple of years later
  - No outside funding
  - Run from revenue from the start
  - Consistently profitable

## How We Present FLOSS

---

- **In a brief introduction, we tend to use the term open source, because it's familiar, and it's not the time to try to distinguish between Free Software and Open Source.**
- **We emphasize convenience and utility to customers**
- **Following is a slide from a (15-slide) introductory presentation to customers**

# The Open Source Advantage



```

-- Main Window : out
-- Home_Dir : Gtk.Window
-- : String)
Signal_Parameters : constant Kernel_Handle
(1 .. 2 | 4 => (1 => GType.Window.Gtk_Win
3 | 5 => (1 => GType.pg)
Handler : Glide_Language_Handle
Handle := new Kernel_Handle_Rece
Handle.Main_Window := Main_Window
Handle.Home_Dir := new String' (H
Glib.Object.Initialize (Handle);
Initialize_Class_Record
(Handle, Signals, Kernel_Class_Handle_Record,
-- Create the language handler
-- so that the embedded gvd uses
-- to guess the language for a
Gtk_New (Handler);
Handle.Lang_Handler := Language_Kernel_Class, "
Glide_Window (Handle.Main_Win
Handle.Lang_Handler;
Handle.Project := Create
Handle.Project_Is_Default
Handle.Predefined_Source_Handler
Handle.Predefined_Source_Handler
Glide_Window (Handle.Main_Window).Lang
Handle.Lang_Handler;
Handle.Project := Create_Default
Handle.Project_Is_Default :=
Handle.Predefined_Source_Handler
Handle.Predefined_Source_Handler

```

- ☺ Source code included
- ☺ Locks-free
- ☺ No limit on number of installs



## Our Pricing Model

---

- **We use a subscription model**
- **Constant annual payment**
- **Typically start with a one year contract**
- **Renewed yearly if customer wants**
- **Licenses are perpetual so no lock-in**
- **Customer free to abandon support if they want to**
- **We specifically avoid the pay-up-front (PUF) model**
- **Over a very long term, we are often a fairly expensive solution, but customer is not committed to this in advance.**
- **We also offer special long term baselining deals**

## Advantages of Pricing Model

- **Customer interests and AdaCore interests are aligned**
- **We make money if customer renews**
- **It is in the AdaCore interest to make sure we are so valuable that the customer does renew**
- **If we sign up a customer and there are no support requests, alarms sound**
- **We find that customers need educating on the value of support. They tend to assume it is useless from experience in the PUF (AKA take the money and run) world, where companies are not really interested in support.**
- **For us, support is the main focus. Not the only focus, we are not selling just support. But a main focus**

## How Does Our Support Work

- **We don't have a support staff!**
- **All our developers have support as their major focus**
- **Customers mostly interact by email and by using our specialized Web interface, GNATracker**
- **Customers report bugs. Sometimes they aren't bugs at all, but if they are we provide work arounds, and if necessary new patch releases (wave fronts)**
- **But the majority of tickets are not about bugs, they are about how to use Ada, how to interpret an error message, how to accomplish specific goals etc.**
- **We are dedicated to rapid response (as CEO I personally oversee the status of all tickets, which shows the importance we attach to this).**
- **Number of tickets opened since beginning of time 75000**

## Our Test Suite

- **Compilers and tools are relatively easy to test**
- **We accumulate a big test suite over time**
- **Mostly from company confidential code**
- **At this stage, we have tens of thousands of test cases, and tens of millions of lines of test code**
- **Mostly highly proprietary and confidential**
- **It's the one part of our system that is proprietary and confidential**
- **We develop small tests that can be shared, but the majority of our test suite cannot be shared.**
- **No changes can be made to core technology without running our entire test suite (side note: used to take one hour, but now with multicore systems, 10 minutes)**

## Our Licensing Approach

- **All tools are GPL-3**
- **GNAT Pro itself is a front end for gcc**
- **So we benefit from the huge amount of work on gcc**
- **In particular from its support for many targets (we support more configurations of GNAT than we have engineers).**
- **All our run-time libraries are modified GPL that permit unrestricted use in executables**
- **Our customers are not generating free software!**
- **Other free licenses used as appropriate (e.g. for our ECLIPSE interfacing work)**

## Our Relationship with the FSF

- **We work closely with the FSF and the GCC and GDB development teams (e.g. Joel Brobecker is release manager for GDB)**
- **GNAT is an Ada front end for GCC, so naturally we depend on all the work done on GCC, particularly the support for multiple targets (e.g. ia64). We actually have more ports of GNAT than we do developers.**
- **We contribute back our updates and modifications to the Ada front end, and to the GCC backend, and to GDB.**
- **Not all our tools are at the FSF (some are just too specialized or relate to very proprietary stuff, or to certification stuff, the latter being better placed at open-do.org).**
- **GNAT Pro releases are based on a stabilized version of GCC with typically fairly extensive patches for GNAT.**

## The Libre Site and the GPL Version

- **The AdaCore Libre site is dedicated to free software development.**
- **Aimed at the academic community, the hobbyist community, and free software developers.**
- **We provide full free support for the academic community.**
- **GAP (GNAT Academic Program)**
  - Over 200 universities enrolled that are using Ada in the curriculum
  - Lots of interesting projects, e.g. the CubeSat program which will eventually orbit and Ada programmed satellite around the moon.
- **Pure GPL Licensed**
  - Dual licensing scheme used
  - GPL version not suitable for proprietary projects

## So Couldn't People Get the FSF Version

- **They could, and it's easy to do so, since it comes packaged with many free GNU/Linux distributions.**
- **But for commercial use, support is critical**
- **And also, warranties and clear licensing are critical**
- **Many companies restrict use of freely downloaded stuff from the net (probably a good idea!)**
- **So in practice almost all commercial development with GNAT is done using supported GNAT Pro from us.**
- **The support is easily worth the price of admission**



## Commercial vs Open Source

---

- **This is a bogus juxtaposition**
- **In every sense, we are a commercial operation**
- **You can contrast Proprietary vs FLOSS**
- **But commercial means engaged in commerce**
- **And That's what we do!**
- **Still huge confusion is out there on what open source and free software means**
- **Following slides show how we present Open Source and Free Software from a purely commercial point of view (not much ideology here!)**

## Open Source

---

- **Open Source is a relatively new term**
- **It refers to the notion of freely providing sources and encouraging a wide community to participate in development**
- **Attractive to major companies (gee, maybe we can get people to do stuff free for us ☺)**
- **Claims are made for better quality, better security etc.**
- **In practice, some OS projects work, some don't. Some OS software is high quality, some is not.**

## Free Software

---

- **A term coined a long time ago by Richard Stallman of the Free Software Foundation.**
- **Refers to the notion of ensuring that the user/recipient of software has freedom to do what they like with it.**
- **Free = Free as in freedom, not free as in free lunch (libre, not gratuit).**
- **The emphasis is on this freedom for users.**

## Open Systems

---

- **An Open System is one built from components with well defined open interfaces.**
- **Free from patent or copyright encumbered formats and interfaces (e.g. no wmf files)**
- **No secret interfaces (e.g. NT kernel)**
- **The idea is that anyone can plug in anything at any point, by adhering to these interfaces.**

## Open Systems, Reality Check

- **A real open system is one that adheres to the philosophy of openness throughout. For example GNU/Linux systems.**
- **Just because you can pass POSIX tests does not mean you are open (e.g. POSIX subsystem for NT)**
- **Just because you have some open components does not make a real open system (e.g. MAC OS X).**

## Free Software vs Open Source

---

- **One of the important freedoms for Free Software is the freedom to modify, which means that sources are available.**
- **So it is often, but not always, the case that FS ends up with an open source community participating in development.**
- **But not all OS projects are free software because licenses may be too restrictive.**

## OS vs FS. What's Important to You?

- **Open Source is not an important issue for the user. In fact there are plusses and minuses in open source:**
  - Pluses
    - May be higher quality due to large developer community
    - May be more secure due to many eyes
  - Minuses
    - Large open developer communities are hard to control. Out of control development is not good.
    - Quality control may not be a major focus of a community of enthusiastic developers.
    - May be less secure due to many eyes

## OS vs FS: What's Important for You?

---

- **Free Software**

- The emphasis is on liberal license terms
- That means that you as a user can do more and have fewer restrictions
- That's an unqualified plus (well we will qualify this a bit later)

- **What about quality?**

- FS does not guarantee quality
- You will have to use your normal procedures to make sure that you choose quality software!
- There is junk/excellent FLOSS
- There is junk/excellent proprietary software



## Free Software Licensing

---

- **Microsoft sells exclusively proprietary programs with a restrictive license**
- **AdaCore sells programs covered by a Free Software License (GPL and GMGPL)**
- **So that's completely different, right?**
- **Actually, there is much more similarity than you might think.**

# Free vs Proprietary Software

- **Microsoft and AdaCore both:**
  - Sell software with support
  - The software is copyrighted, and cannot be copied or used without a license.
  - Provide a license that allows you limited rights to copy the software and use it.
  - You pay for the combination of the software, support, and appropriate license.
- **So what is the difference?**

## **It's All in the License Terms**

---

- **The ONLY difference between MS and AdaCore is the terms of the license**
- **But it's quite a difference because the license conditions are very different.**
- **The MS license is all about strictly limiting the usage and copying.**
- **The AdaCore license is far more liberal and has very few restrictions.**

## Differences

---

- **Anything the MS license allows is also allowed by the AdaCore License**
- **But the AdaCore (GPL/GMGPL) is much more permissive**
  - Unlimited copying allowed
  - Unlimited use
  - Unlimited distribution
  - Full sources available
  - You can modify and distribute sources
  - Run-time library can be modified and used freely in any of your programs in any way you like.

## So What Can't I Do

---

- **The GPL/GMGPL allows a lot but it does stop you from doing some things.**
- **You can modify the software (e.g. the Ada 95 compiler, GNAT) and use it freely**
- **You can even distribute the modified version.**
- **But if you do, you must use the GPL for that distribution and include sources including your modifications.**

## GPL: A Virus?

---

- **I have heard that the GPL is a virus which will cause me to lose all my rights to my own code.**
- **Yes, there are some parties that have been energetically spreading this peculiar rumor.**
- **Let's look at the facts and you decide for yourself if Virus is an appropriate term 😊**

## The GPL Virus in Action ???

---

- **Suppose you:**
  - Obtain the sources
  - Modify them
  - Distribute the modified version
- **What do the two licenses have to say about this sequence of events?**

# The Microsoft Response

- **Obtain the sources**
  - Where did you get them? Our lawyers want to know. That's stolen property, cease and desist or else.
- **Modify them**
  - You are creating an illegal derivative work. Enclosed is a notice of a suit we are filing for copyright infringement. You have no rights to do this.
- **Distribute the modified version**
  - This is getting more serious, the statutory penalty for such distributions, which violate our copyright, is \$50,000 for each distributed copy.



# The AdaCore Response

---

- **Obtain the sources**

- We hope you didn't have any trouble locating the sources. Let us know if you did and we will help you out. The documentation contains full details on how to obtain and build them

- **Modify them**

- You are welcome to do so. We can provide some help under your standard support contract, but if you make major modifications, then this may limit the support we can provide. We are happy to discuss special support arrangements. Also you might want to discuss your modifications with us. Perhaps it is something we would be happy to do. Perhaps you would like to contribute your modifications so they will be in future releases.

- **Distribute the modified version**

- No problem. The only rule is that you need to distribute the full sources (our original stuff with your modifications), and use the same license (the GPL) for everything. You don't have to distribute, and as long as you don't you are welcome to keep your modifications secret.

## The Virus Effect, Part 2

---

- **OK, I understand about modifying the compiler, and that's fine, I'm not in the compiler modification business anyway.**
- **But what about the run time, don't my programs include some of your run-time?**
- **Answer: yes, they do. If the run-time was under the GPL, there would be a problem.**

## A GPL'ed Library or Run-Time

- **Even if you don't modify, your program that includes such a library would be covered by the GPL.**
- **That's fine for people who want to write Free Software**
- **But not so fine at all for people who want to write non-free software, e.g. classified weapons system software.**

## Worrying about Licenses

- **Free Software and Proprietary Software share an important common truth:**
- **CHECK THE LICENSE!**
- **Make sure it is suitable for your use**
- **A GPL'ed library (e.g. CYGWIN from Redhat) is not suitable for use in non-free software.**
- **So you need to look elsewhere (e.g. Redhat sells a separate version of CYGWIN with different more suitable licensing).**

## What About the AdaCore License?

- **We know our compiler and tools are going to be used in non-free-software contexts.**
- **After all a great deal of the Ada business is in defense systems for example.**
- **So we make very sure that our licensing is appropriate for this use.**
- **And we provide a simple and clear license statement as part of the product that guarantees that this is the case.**

## The GPL and the GMGPL

- **The GMGPL is used in GNAT for ALL components which might be included in your program:**
  - Run Time Library
  - General use libraries
  - Generic code you need to instantiation
- **The GMGPL specifically allows you to incorporate this in your program without any effect on the IPR status. The program can be proprietary, classified etc completely at your discretion.**
- **CHECK THE LICENSE!**

## **FLOSS Dangers**

---

- **Automatic loss of IP, forced source distribution NOT!**
- **Use of unsupported and unreliable software**
- **Improper inclusion of GPL software in products**
- **Risks in distributing GPL software**

## Automatic Loss of IP NOT!

- **If I include GPL software deliberately or accidentally, won't I be forced to distribute my sources under the GPL?**
- **ABSOLUTELY NOT!**
- **You can offer a GPL to someone if you wish, but it can never happen automatically.**
- **If you include GPL software in proprietary products, it is likely a copyright infringement just as it would be if you included Microsoft proprietary software.**
- **You may be subject to statutory and damage penalties**
- **You will have to cure the situation, usually by removing the offending software**
- **With the GPL, you *may*, but are not *required*, to cure the situation by changing your licensing to the GPL.**



## Forced Distribution of Sources

- This can *NEVER* occur
- As per the previous slide, the GPL gives you an option to cure a copyright infringement not available with proprietary source.
- You may *at your choice and discretion* change your licensing to the GPL and distribute your sources.
- But you do not have to exercise this option, and almost certainly in typical situations of proprietary or classified software distribution, you will not exercise this option.
- So the GPL doesn't offer an advantage in practice, but it's also not the case that the GPL has any disadvantages here. You are no worse off than you would be with proprietary licenses.

## Use of unsupported and unreliable software

- **To many, open source is synonymous with stuff you can download free from the internet.**
- **Such software is often of dubious provenance, and comes with no support.**
- **And you have no guarantee of correct licensing (headers in files have no legal significance)**
- **There is no/minimal warranty, who's there anyway?.**
- **And fitness for purpose, reliability etc are unguaranteed**
- **All in all, not the sort of stuff you want around?**
- **This is a matter of internal policy. Many companies have policies that severely restrict or control downloading of unsupported software. Such policies are reasonable and often advisable. Some companies allow FLOSS only if it is commercial and fully supported.**

## **Improper inclusion of GPL software**

- **To uneducated programmers, the free in Free Software sounds like they are free to do anything they want.**
- **Even though they know they should not include proprietary software from Oracle, Microsoft etc in distributed products, they think it is fine to include “free” software.**
- **They are wrong, and this mistake can lead to costly and embarrassing copyright violations.**
- **An important task is to educate programmers at all levels to understand that they need to regard all software as potentially restricted and copyrighted.**
- **Software should only be copied from other sources after very careful checking of the licensing conditions (and remember headers on files mean nothing!)**

## Risks in distributing GPL software

- **You are allowed to redistribute GPL'ed software provided you follow the rules (include sources etc)**
- **But just because a license allows you to do something does not mean you have to do it.**
- **And just because a license allows you to do something does not mean it is a good idea.**
- **Distributing other people's software is potentially risky. You can acquire legal liabilities that you definitely do not want.**
- **You wouldn't consider routine redistribution of Microsoft software.**
- **So why consider redistributing FLOSS?**
- **It's probably a bad idea, and most likely, you should have a policy preventing it.**

## Warranties

- **Basically the issue of warranties is completely orthogonal with FLOSS vs proprietary licensing.**
- **There can be FLOSS with extensive warranties, or proprietary software with minimal warranties (most mass market software).**
- **Standard freely downloaded software under the GPL has minimal warranties, and the GPL specifically disclaims warranties. But companies are free to provide warranties if they choose to.**
- **As with any acquisition, you need to read licensing and warranty documents carefully and make sure they are suitable for your needs.**
- **Commercial FLOSS typically has stronger warranties than freely downloaded stuff, but the industry is not in general in the business of strong warranties.**

## FLOSS - Conclusion

- **FLOSS licenses like the GPL give you many rights**
  - Modify the program
  - Redistribute modifications
  - Include software in your products if you redistribute using the GPL
  - Redistribute original program
- **But just because the license gives you freedoms does not mean you have to take advantage of them.**
  - It is perfectly permissible for a company to have restrictive policies
  - That restrict or entirely prevent the exercise of some of these freedoms.
  - The GPL does not stop you from doing that!
- **Education is important**
  - There is lots of confusion when it comes to FLOSS
  - Not surprising when big companies (Microsoft, GHS, SCO etc) have an interest in promoting that confusion.
  - FLOSS is in wide use (particularly GNU/Linux), and it's important to fully educate your technical staff on limitations, policies, appropriate use etc.

## Bottom Line

---

- **AdaCore is a successful commercial company that shows that FLOSS can work in a commercial setting.**
- **People keep asking us how we make a living. Answer is we do it the same way any other software company does, by selling high quality software at a reasonable price.**
- **But couldn't people redistribute all your stuff**
  - They could, but they don't in practice, why would they
  - Even if they did, it would not come with our guarantees and support
  - So that in practice is not an issue

## Would this Approach Work In Other Cases

- **Depends, seems like it works best where there is a significant need for support.**
- **Could MS open source Windows**
  - Probably, it's an interesting thought
- **Could a company making games open source them**
  - Dubious, games cost millions of dollars, and the copyright prevention of copying is a key element in making the low cost, wide distribution model work successfully.
- **We have been successful in convincing other companies to use FLOSS for their products**
  - Praxis and SPARK
  - Sofcheck (now part of AdaCore!) and CodePeer
- **We think the model could be used much more widely**