



# **goanna *studio***

---

QUICK START GUIDE – VERSION 2.1  
COMMAND LINE EDITION

---

Copyright © 2008-2010 NICTA  
All rights reserved.

---

This document, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information contained herein is the property of NICTA and is confidential between NICTA and the client and remains the exclusive property of NICTA. No part of this documentation may be copied, translated, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of NICTA.

NICTA does not warrant that this document is error-free.

---

NICTA Corporate Head Office  
Australian Technology Park  
Level 5, 13 Garden Street  
Eveleigh NSW 2015  
Australia

Web: <http://www.redlizards.com>  
Support: [support@redlizards.com](mailto:support@redlizards.com)

---

## INTRODUCTION

Only a few simple steps are required to get Goanna running on your local machine. This document will guide you through the process of installing Goanna from the binary distribution package, and using it to check some provided example code.

Goanna runs on 32-bit and 64-bit Linux distributions for many of the recent gcc compiler version. The following installation instructions describe an installation on a Debian system, but installation on other systems should be fairly similar.

## INSTALLATION



Quick solution: Unpack the Goanna distribution, run `./install-goanna` and follow the instructions. Don't forget to set the environment variables as instructed.

For the installation of Goanna follow the next few steps:

1. Get the binary distribution package and place it in your home directory. The package is typically named `goanna-osname-number.tgz`, where *osname* is the name of the OS and *number* is the revision number.
2. Unpack the package with: `tar -zxvf goanna-osname-number.tgz`
3. Go into the directory just created (the name is `goanna-number`). Then, start the install script:

```
./install-goanna
```

When asked about the directory to install the binaries, select one that suits your needs. We suggest installing it in a directory in your home directory, e.g. `$HOME/goanna`. We will call this directory the Goanna installation directory from now on.

The install script will check for a number of libraries that need to be installed on your machine and will search for your standard compilers, i.e. `gcc` and `g++`, and detect a few more things that Goanna needs to know in order to work. It will also install a number of files in your Goanna installation directory.

4. Set your `PATH` environment variable to include the Goanna installation directory, so that you can easily start Goanna, e.g. with this command (for `bash` users):

```
export PATH=$PATH:$HOME/goanna/bin
```

Where `$HOME/goanna` is your Goanna installation directory. You can also add this to your shell's initialization file, e.g. to `$HOME/.bashrc` if you are using `bash`.

5. Set the `EDG_BASE` environment variable to point to the directory in which you installed Goanna. It is used to find the location of `edgcpfe` and `predefined_macros.txt` which are both needed by `goannacc` and `goannacc++`. Again, in `bash` you can use:

```
export EDG_BASE=$HOME/goanna/bin
```

Where `$HOME/goanna` is your Goanna installation directory. You can also add this to your shell's initialization file, e.g. to `$HOME/.bashrc` if you are using `bash`.

## NON-DEFAULT COMPILER OR CROSS-COMPILER

If you intend to use other compilers than the standard system compilers on your machine, e.g. cross-platform compilers, then you need to configure Goanna so that it uses the right compilers and header files. You can do that by starting the `goannacc` and `goannacc++` scripts respectively, with the `--configure --with-cc <PATH>` arguments, where `<PATH>` is the full path to the compiler executable, e.g. like this:

```
goannacc --configure --with-cc /cross/bin/arm-linux-gcc
```

```
goannacc++ --configure --with-cc /cross/bin/arm-linux-g++
```

This will create configuration files in your home directory (`.goannaccrc` and `.goannacc++rc` respectively), which contain a few variable definitions that are important for Goanna to work correctly with the compilers you selected.

We strongly suggest not to edit these files manually, or at least to edit them only if you really know what you are doing. If you switch to another compiler, start the scripts as described above to tell Goanna about your new compiler.

## RUNNING GOANNA



Quick solution: Use `goannacc` as you would use `gcc`. This runs `gcc` as well as doing all the additional checking as defined in `properties.init` in your installation directory.

**Running on C files:** The easiest way to use Goanna is by using it as a drop-in replacement for your standard compiler such as the `gcc` or `g++` compiler. Just call

```
goannacc filename.c
```

**Running on C projects:** If you want to use Goanna to check existing projects which use `gcc`, then `goannacc` is an easy way to do this without actually having to change any makefiles at all. Set the environment variable `CC` to `goannacc` and call `make -e` to override the `CC` value in the makefile with the environment variable. Of course, you can also replace `gcc` in your makefiles or build scripts with `goannacc` and `g++` with `goannacc++` respectively.

**Running on C projects with autoconf:** For those projects that use autoconf for configuration, the command to use is:

```
CC="goannacc" ./configure
```

This should mean that `goannacc` is used as the C compiler.

**Running on C++ files/projects:** For every project that contains C++ and you typically use the `g++` compiler follow the instructions above but use `goannacc++`.

## PURCHASE AND LICENSE ACTIVATION



To purchase Goanna, go to <http://redlizards.com/purchase.html> and follow the instructions. To activate a license, go to <http://redlizards.com/activate.html> and follow the instructions.

Purchase Goanna by going to the purchase page <http://redlizards.com/purchase.html> and following the purchase instructions.

Once you purchase one or multiple licenses you will receive an email with the details of your purchase, including your order number. This gives you the right to activate a matching number of seats.

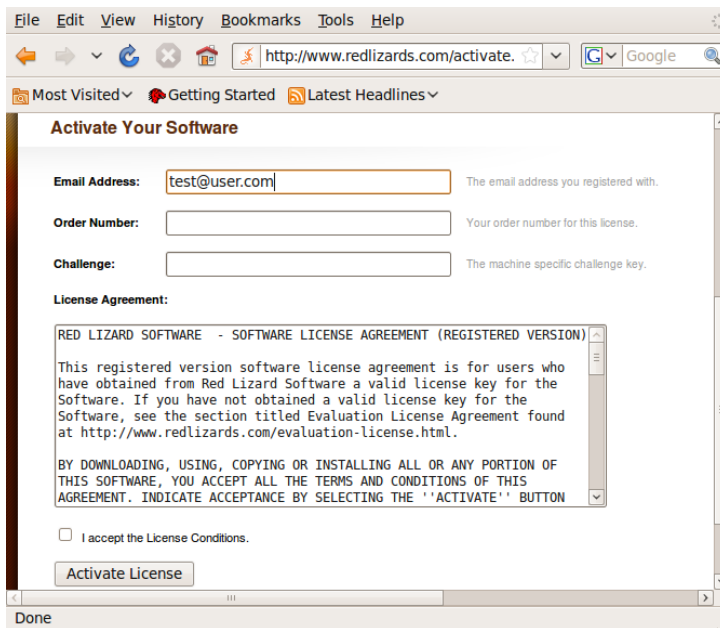
Go to the activation page <http://redlizards.com/activate> and enter the following details:

**Email address:** The email address you provided when purchasing Goanna

**Order number:** The order number provided in the Goanna purchase email

**Challenge:** The challenge key of the computer for which you have purchased a license. This can be obtained by calling:

```
goanna-key
```



The resulting license file, called `goanna_license.lic`, will be emailed to the email address you provided. You will also be taken to a page where you can download the license file by clicking on `Download your license`.

Once you obtain the license file, set `LMX_LICENSE_PATH` to the directory containing the file. For example, you would call:

```
Export LMX_LICENSE_PATH=$HOME/goanna
```

if you placed the license file in `$HOME/goanna`.

## CHECKS



Quick solution: Individual checks can be switched on/off in `properties.init` in your installation directory. Note, by default not all checks are on.

The standard checks available for Goanna are defined in the `properties.init` file in your Goanna installation directory. It is a text file containing entries of the following form:

```
#
# In some executions, a pointer is used after it has been
# freed.
#
MEM-use-free-some
```

Each entry contains a short description of a check together with the handle of the check. The “#” is used as a comment. You can switch off checks by putting the comment symbol # in front of the name of a check. Removing # will switch the check on again.

By default not all checks might be switched on.

The full description of each check can be found in the *Goanna User Guide*.

## COMMANDS AND OPTIONS

Goanna accepts a number commands and options that enable additional analysis techniques and runtime behaviors. The most common ones are listed below. For a full description of all flags see the *Goanna User Manual*.

### FLAGS AND OPTIONS FOR GOANNACC

The following flags and options can be directly passed to `goannacc/goannacc++`. Options define certain runtime behavior of Goanna as well as additional analysis techniques that help to find more bugs more precisely.

<code>--help</code>	Prints all available options.
<code>--ipa</code>	Interprocedural Analysis. Requires the <code>-db=&lt;db&gt;</code> option.
<code>--no-compile</code>	Does not run the compiler at the same time. <b>Note:</b> This requires the program to be <i>syntactically correct</i> as Goanna does not replace the syntax checking of the compiler.
<code>--property-file=&lt;file&gt;</code>	Uses the checks in <code>&lt;file&gt;</code> instead of the <code>properties.init</code> file.
<code>--trace</code>	Prints out a trace through the function that leads to the error warning. This is helpful to for understanding why a bug occurred.
<code>--timeout=&lt;seconds&gt;</code>	This is a per-file time-out. This is useful if you have some files that take very long to analyze and you like to set an upper limit. Goanna will run for <code>&lt;seconds&gt;</code> and terminate with the issues found until then.
<code>--with-cc=&lt;compiler&gt;</code>	Takes the specified <code>&lt;compiler&gt;</code> instead of standard one.

For example, to run Goanna on `myfile.c` without invoking the compiler you call:

```
goannacc --no-compile myfile.c
```

More than one option can be combined. An example is:

```
goannacc --trace --timeout=30 myfile.c
```

The above runs Goanna including Array Bounds Checking and times out after 30 seconds. Of course, options can be included as defaults into makefiles, for example by setting:

```
CC="goannacc --trace --timeout=30"
```

## TROUBLESHOOTING

This section describes a number of known issues and solutions to them. Should you find more issues you believe we should know about, please contact us at [support@redlizards.com](mailto:support@redlizards.com).

### AVOIDING TROUBLE

There are a few simple things that can save you from running into problems right at the start:

1. Make sure you set your `PATH` environment variable correctly.
2. Make sure you set your `EDG_BASE` environment variable correctly.
3. Uninstall any old version of Goanna before installing a new version.

### SOLVING TROUBLE

There are some issues that might occur depending on your exact compiler version, include files, hardware, OS version, undocumented features in C/C++ or a combination of the above. Some known issues are listed below.

#### **PROBLEM: GOANNA IS NOT USING THE RIGHT HEADER FILES.**

Did you tell Goanna about your compiler? If not, then you should run `goannacc` and/or `goannacc++` with the arguments `--configure --with-cc <PATH>`, where `<PATH>` is the full path to your compiler's executable.

---

## PROBLEM: PREDEFINED MACROS

You might get an error such as:

```
Catastrophic error: incompatible redefinition of predefined macro
    "__WCHAR_TYPE__"
```

This typically indicates that your particular machine has predefined data types that are different from what Goanna expects.

To solve this issue go to the `lib` subfolder of the directory where Goanna is installed (e.g., `$HOME/goanna/lib`) and comment out those lines in the `predefined_macros.txt` file, which refer to the error. In the above case it requires commenting out one line as follows:

```
# gnu no  __WCHAR_TYPE__ int
```

Save the modified `predefined_macros.txt` file and run Goanna again.

---

## PROBLEM: UNSUPPORTED BUILT-INS

Occasionally, you might see an error such as:

```
identifier "__builtin_offsetof" is undefined
```

Compilers use a large number of C/C++ language extensions. Some are better documented than others, some are for internal purposes only, but still end up in code. Goanna supports as many built-in functions as possible, but every now and then we come across new ones and they are actually hard to integrate.

Feel free to let us know about the ones that give you pain. We are working on it!

---

## PROBLEM: NUSMV CANNOT FIND EXPAT

When you run the version of NuSMV that comes with Goanna, you might get the following:

```
error while loading shared libraries: libexpat.so.1: cannot
open shared object file: No such file or directory
```

In this case have an old version of the expat XML parser installed, or no version at all. You can obtain a recent version of expat at sourceforge <http://expat.sourceforge.net/>, which should solve the problem.