



Quick setup of your Narval assistant

Quick setup of your Narval assistant

Sylvain Thénault

Edition: Version 1.5 - 2004-07-06

Table of Content

Chapter I - Software installation	1
1. Using a package	1
2. Using the source distribution	1
Chapter II - Setup of your assistant	2
1. Creating a personnal assistant	2
2. What do I find in a narval home ?	2
Chapter III - Running your assistant	3
Chapter IV - Using Horn	4
Chapter V - Stopping your assistant	5
Chapter VI - Comments, support, bug reports	6

Chapter I

Software installation

1. Using a package

According to your platform, you should be able to install narval using available RPM or DEB packages. This is the easiest way to install it since it will automatically handle dependencies for you.

2. Using the source distribution

First you'll have to install Narval's dependencies : * a python \geq 2.2 interpreter, of course (<http://www.python.org> [<http://www.python.org>]) * logilab's common library (<http://www.logilab.org/projects/common> [<http://www.logilab.org/projects/common>]) * twisted (<http://www.twistedmatrix.com/> [<http://www.twistedmatrix.com/>])

If you want to use Horn (narval's graphical user interface), you'll also have to install: * pyro (<http://pyro.sf.net> [<http://pyro.sf.net>]) * pygtk2 (<http://www.pygtk.org/> [<http://www.pygtk.org/>]) * xmltools (<http://www.logilab.org/projects/xmltools> [<http://www.logilab.org/projects/xmltools>])

Some optional packages are also needed to use some actions modules, such as:

- PyLogDB requires pylog (<http://christophe.delord.free.fr/en/pylog> [<http://christophe.delord.free.fr/en/pylog>])
- Metalog requires pymetalog (ftp://ftp.logilab.org/debian/sid/pymetalog_2.0.b.orig.tar.gz [ftp://ftp.logilab.org/debian/sid/pymetalog_2.0.b.orig.tar.gz])
- RedLand requires the python bindings for the Redland RDF library (<http://librdf.org/> [<http://librdf.org/>])

To install Narval, extract the source code from the tarball, go to the narval-X.Y directory and type:

This will install both narval. Do the same thing with the horn package to install it if necessary. You can type:

to see available installation options. That maybe usefull if you want to install it without being root for instance. Notice that in this case you'll probably have to set the NARVAL_SHARE environment variable to provide to narval the location of its shared resources.

Once the installation done, you should have the following executable installed:

- *narval*, the narval interpreter
- *npm*, the narval package manager
- *narval-runtest*, a developpment tool used to run functional tests
- *horn*, the narval graphical user interface (if the separated package has been installed)

Chapter II

Setup of your assistant

1. Creating a personnal assistant

A narval assistant is tied to a "narval home". You can create a narval home, and so an assistant by launching the narval interpreter with the `--create-home` option:

This will initialize a fresh narval home or add missing files to an existant one. By default, the narval home is the `.narval` directory in your home directory. You can also specify an alternate home using the `NARVALHOME` environment variable or using the `--home` option.

Of course you can have multiple homes to run multiple assistants, optionally dedicated to one or more particular tasks.

2. What do I find in a narval home ?

You'll see the following directories under a narval home:

`data`
a special directory to store miscellaneous data

`recipes`
contains user specific xml files defining recipes

For now, you'll be mainly interested in two files:

- **data/memory.xml**, the initial narval's memory, loaded on narval startup. You can configure the behaviour of your narval by modifying this file or better, files included by this one (this will probably ease the process of future upgrades). NOTICE: due to a bug in PyXML, you may have to edit the original memory.xml file to replace relative path to included files with absolute path.
- **data/narvalrc.xml**, the configuration file indicating what should / should not be loaded from the narval home.

Chapter III

Running your assistant

To start your assistant, you usually just have to type:

The behaviour of your assistant will now be driven according to its initial memory file, which tell it which communication channel to listen on, which recipe to start and so on. You can also directly interact with it using horn. However to do so, you have to start narval with the pyro server activated :

For now, most interactions between the agent and its user is done via the jabber protocol, since it has been used as communication channel to implement the first stories of the project. To see how to configure the jabber protocol handler, which recipes are available and what can you do with them, take a look a the [Narvabber how-to](#) [jabber_bot.html] .

Chapter IV

Using Horn

You can use the narval's graphical front end to :

- edit the content of your narval home (actually only recipes may be edited from the gui)
- observe / interact with your assistant (you'll have to start narval with the pyro communication service enabled for this: `--pyro-port=XXXX`)

To start horn, just type:

You will then be able to use the "Narval -> Connect" menu from Horn to connect to Narval (using your hostname and 8787 as the port and pyro as the method). Play with the menus if you want, but don't lose too much of your time and... read the tutorial at <http://www.logilab.org/projects/narval/documentation/tutorial> [http://www.logilab.org/projects/narval/documentation/tutorial] that describes how to download, set up and start an existing application available from <ftp://ftp.logilab.org/pub/narval/applications> [ftp://ftp.logilab.org/pub/narval/applications].

But be aware that you most probably won't get away without reading the fine manuals :-).

Chapter V

Stopping your assistant

If you want or have to stop your narval assistant, you can either:

- just type "narval --stop" on the command line (you may have to use the NARVALHOME environment variable or --home option to locate the correct narval home)
- type Ctrl-Z and then "kill %" on the terminal where you've started narval (Ctrl-C is actually not working due to a problem with python thread support and signal interaction)
- use horn, connect it to your narval and use the shutdown action from the menu
- just tell him "die" using the jabber interface (you must have a configuration allowing this, see the jabber chatbot how-to)

Chapter VI

Comments, support, bug reports

Use the ai-projects@logilab.org [mailto:ai-projects@logilab.org] mailing list. Since we do not have publicly available bug tracker yet, bug reports should be emailed there too.

You can subscribe to this mailing list at http://www.logilab.org/maillinglists/ai_projects/maillinglist_register_form
[http://www.logilab.org/maillinglists/ai_projects/maillinglist_register_form]

Archives are available at <http://lists.logilab.org/pipermail/ai-projects/> [http://lists.logilab.org/pipermail/ai-projects/]