

## Introduction

We can build RPMs for the Ganga packages. This is currently a manual operation, performed by calling `~/ganga/release/tools/rpm_builder.py` with the `gangage` account, but could ultimately be included during step 9 of the Ganga release procedure described here. As of April 2013, the RPMs are known to install successfully, but require further testing before being considered production-ready. We make use of the Python Distutils [package](#).

*Note that the `rpm_builder.py` script needs to be run from an SLC6 node. lxplus nodes should become exclusively SLC6 from 6th May 2013. There is now also a GangaSuite package available which contains the entire Ganga codebase. This still needs a bit of work doing*

`~/ .rpmmacros`

We need the following defined in `~/ganga/.rpmmacros`, otherwise RPMs will not be built.

```
%__os_install_post %{nil}
%_unpackaged_files_terminate_build 0
%_prefix /opt/ganga/install
%debug_package %{nil}
%_binaries_in_noarch_packages_terminate_build 0
```

### `rpm_builder.py` script

`release/tools/rpm_builder.py` is used to build the RPMs. Here's a quick summary of what it does when called with the `-v 6.0.0` command line option:

- Export (not checkout) the specified Ganga release (not pre-release) from `svn+ssh://svn.cern.ch/repos/ganga/tags/`. Note that specifying `-v 6.0.0` implies the existence of the SVN tag `Ganga-6-0-0` in SVN. The release will be exported to `/tmp/gangabuild` or whatever directory is passed to the `-b/--builddir` option.
- Create the RPM pre/post install/uninstall files (these get executed during RPM installation/removal).
- Create the Distutils manifest template (equates to 'include all files').
- Since Ganga's layout is not entirely compatible with the assumptions Distutils makes, we have to create a temporary `python/GangaBin` directory containing the release, templates, bin directories and the `LICENSE_GPL` file. These are subsequently packaged in the `GangaBin-6.0.0-1.noarch.rpm` file.
- For each Ganga package (e.g. `GangaBin`, `Ganga`, `GangaAtlas`, `GangaLHCb` etc):
  - ◆ Write the `setup.cfg` and `setup.py` files required by Distutils.
  - ◆ Execute the distutils build command.
  - ◆ Record success/failure
  - ◆ Clean up RPM build workspace directory
- Print a build summary:

```
#####
#####
##### Build Summary #####
#####
#####
GangaCamtology:      OK
GangaSuperB:         OK
GangaPanda:          OK
GangaAtlas:          OK
GangaDirac:          OK
GangaSAGA:           OK
GangaPlotter:        OK
GangaService:        OK
GangaRobot:          OK
```

## GangaRPM < ArdaGrid < TWiki

```
GangaTest:          OK
  GangaCMS:         OK
GangaTutorial:     OK
  GangaLHCb:       OK
  GangaBin:        OK
GangaGaudi:       OK
  Ganga:          OK
```

Within `rpm_builder.py` we use a set of python dictionaries to configure attributes that will end up in the RPM file. For example, we specify the RPM requirements of the Ganga packages with:

```
rpm_require_map = {
'GangaBin' : "python >= 2.4.3, Ganga >= "+this_version,
'Ganga' : "GangaBin >= "+this_version,
'GangaAtlas' : "Ganga >= "+this_version,
'GangaCamtology' : "Ganga >= "+this_version,
'GangaCMS' : "Ganga >= "+this_version,
...
}
```

The effect of this is that `yum install GangaAtlas` will automatically install Ganga, GangaBin and Python 2.4.3 (if required). We similarly populate the RPM description attributes:

```
description_map = {
'Ganga' : 'The Core Ganga package',
'GangaBin' : 'Contains the Ganga executable, release scripts, documents and templates',
'GangaAtlas' : 'The Ganga ATLAS package',
...
}
```

### **setup.cfg file (auto-generated by rpm\_builder.py)**

Settings of note defined in this file include:

```
[bdist_rpm]
dist-dir = /afs/cern.ch/sw/ganga/www/download/repo/NOARCH
vendor = "Ganga <project-ganga-developers@cern.ch>"
requires = Ganga >= 6.0.0

[install]
prefix = /opt/ganga/install/python
```

The `dist-dir` defines where the newly created RPMs are placed, `prefix` can be overridden by the end-user to define a custom install location:

```
[root@dashb-ganga32 ~]# rpm --nodeps -ivh --prefix /tmp/junk /afs/cern.ch/sw/ganga/www/download/repo/NOARCH/GangaAtlas-6.0.0-1.noarch.rpm
Preparing...      ##### [100%]
 1:Ganga          ##### [100%]
[root@dashb-ganga32 ~]# ls /tmp/junk/python/Ganga/
CLIP  GPI    __init__.py  PACKAGE.py  scripts  Utility
Core  GPIDev  Lib          Runtime     test
```

### **setup.py file (auto-generated by rpm\_builder.py)**

This file calls the `Distutils setup()` method, and passes a few pieces of info, such as the package name, version number, author info and text descriptions of the package:

```
name = "GangaDirac",
```

```
version = "6.0.0",

###DESCRIPTION###
description = 'Description goes here',

###LONG_DESCRIPTION###
long_description = "Long description goes here",

url = "http://ganga.web.cern.ch/ganga/",
author = "The Ganga Project",
author_email = "project-ganga-developers@cern.ch"
```

## Yum repository

You can either directly install the RPMs from their location at

`/afs/cern.ch/sw/ganga/www/download/repo/NOARCH` or configure your Yum client to know about the repository (which is auto-generated by `rpm_builder.py`. Create a file at `/etc/yum.repos.d/ganga.repo` containing:

```
[Ganga]
name=Ganga repo
baseurl=http://ganga.web.cern.ch/ganga/download/repo/NOARCH/
enabled=1
gpgcheck=0
```

## Future developments

- Handle hotfix releases properly (including the internal RPM release version attributes being sane).
- At the moment we build the RPMs with the `--no-autoreq` option, which prevents the automatic calculation of RPM requirements for files within Ganga. For example, some files require Perl/C libraries. Thus we assume that the installation is being performed on a node with these available (e.g. a Grid user interface).
- External dependencies.
- Populate all RPM metadata attributes using python dictionaries.
- Figure out exactly which of these [licences](#) we should be using.
- Create an RPM that installs everything. **Done: we create the GangaSuite package for this**

---

This topic: ArdaGrid > GangaRPM

Topic revision: r6 - 2013-05-03 - MikeKenyon



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback