Table of Contents

A prepare method for Ganga applications

This documents the 'prepared' state for Ganga Core applications, as introduced in Ganga 5.7.0.

High-level overview

The purpose of the prepared state is to allow users to 'freeze' an application in a known state, such that the exact same analysis job can be executed in the future (optionally over different input data). During the prepare phase, files that are integral to the application's execution (such as custom binaries or experiment-specific software areas) are copied to the user's Shared Directory (ShareDir) which is by default <gangadir>/shared/<user>.

This functionality already existed in the Athena backend, as exposed by the prepare() method, but the resulting configuration files were stored in the /tmp directory, the persistence of which cannot be predicted. Similar underlying techniques will be exploited to allow initially Executable (and later Root) applications, plus any defined within the GangaTutorial package, to be placed into a prepared state.

Demonstration - Executable() application

We use an attribute attached to the application to indicate whether it has been prepared:

```
Out[23]: Executable (
 exe = 'echo' ,
 env = {} ,
 args = [Hello World] ,
 is_prepared = None
 )
```

The **`is_prepared`** attribute will hold a ShareDir object, which is generated with a random name in the users gangadir/shared directory:

```
Out[27]: ShareDir (
 name = '/home/mkenyon/gangadir/shared/mkenyon/conf-41540084-7372-1309946661-72' ,
 subdir = '.'
 )
```

As an example, we can configure the default Ganga job, which has the above Executable() application attached to it:

```
In [1]:a=Job()
In [2]:a.prepare()
Ganga.GPIDev.Schema                : INFO     Preparing Executable application.
Ganga.GPIDev.Schema                : INFO     Created shared directory: /home/gangadir/shared/con
```

Note that it is equivalent to prepare the job, or the application associated with the job. In other words

```
job.prepare()
```

and

```
job.application.prepare()
```

are equivalent. Additionally, submitting a job also automatically calls the prepare method behind-the-scenes. The result of running the prepare phase is a job with the following application attributes.

```
Out[10]: Executable (
 exe = 'echo' ,
 env = {} ,
 args = [Hello World] ,
 is_prepared = ShareDir(name='/home/gangadir/shared/conf-d7533df3-066a-4922-81ca-ec6404a10c6a',su
 )
```

The contents of the ShareDir depend on the type of application that was prepared. In the basic example above, the application would attempt to execute the command 'echo' on the backend/workernode, so we don't need to copy anything to the ShareDir. In a more realistic case, though, we might have a custom script that we wish to run on the worker node. This would then be copied to the ShareDir during the prepare phase.

Once an application has been prepared, it gains a reference counter which is stored in the Ganga metadata system, and can be checked by calling shareref:

```
In [10]:shareref
Out[10]:
                                Shared directory |         Date created |  Reference count
 ----------------------------------------------- | -------------------- |  ---------------
                conf-12754112-8042-1314361528-33 | 26 Aug 2011 13:25:28 |                1
                   conf-86403407-38-1314361595-8 | 26 Aug 2011 13:26:35 |                1
                conf-24411345-7135-1314361709-42 | 26 Aug 2011 13:28:29 |                1
                conf-69852897-3565-1314541740-23 | 28 Aug 2011 15:29:00 |                1
```

If an application is later associated with another job, or place in the box, the reference counter is incremented. Likewise, it's decremented when a job or box object is removed. ShareDirs with a reference count of 0 will be removed when Ganga next closes down; in the event that an application's ShareDir cannot be found during Ganga closedown, the application will be unprepared.

The contents of the shared directories can be viewed:

```
In [12]:shareref.ls('conf-284cbb3e-da37-4f6c-87e3-55576ec82cb7')
|  conf-284cbb3e-da37-4f6c-87e3-55576ec82cb7/
|  |  runMain.C
```

Applications not associated with a persisted Ganga object (such as a job or the box) can be prepared, but they, and their associated ShareDir, will not persist beyond the current Ganga session.

Copying a (prepared) application

Copying a prepared application/job object results in an identical copy of that object (ie. referencing the same ShareDir) which will therefore have some of its attributes set read-only. It is possible that the user may wish to modify these attributes. This can be achieved by passing the **`unprepare`** argument to the copy method:

```
a=Job()
a.prepare()
b=a.copy(unprepare=True)
```

Note that by default (ie without the **`unprepare`** argument), calling **`copy()`** will not unprepare the application. It is possible to modify the default behaviour of the copy method such that an application will be unprepared when copied. This can be set either temporarily from the Ganga command line:

```
config['Preparable']['unprepare_on_copy']=True
```

or permanently, by adding the following stanza to **`~/.gangarc`**:

```
[Preparable]
unprepare_on_copy = True
```

Unpreparing an application

Once an application has been prepared, some of its attributes (as determined by that application's developers) will become read-only. This is to prevent the user accidentally changing an attribute on a prepared application (and hence violating the sense of a prepared application). In the event that the user really does want to modify a procted attribute, they should either unprepare the application/job

```
j.app.unprepare()
app.unprepare()
```

or copy the application/job to a new instance in the following manner:

```
newjob=oldjob.copy(unprepare=True)
```

Applications can also be unprepared by resetting their `is_prepared` attribute to `None`:

```
In [1]:a.application.is_prepared=None
Ganga.GPIDev.Base.Proxy            : INFO     Unpreparing application.
```

Shareref table bookkeeping

The shareref table (see above) has to be kept in sync with the prepared jobs/applications in the repositories, and the ShareDir directories on disk. To achieve this, a method is made available through the GPI:

```
In [2]:shareref.rebuild()
```

which allows the user to completely rebuild the table, in the event that it becomes inconsistent with jobs and/or ShareDir directories. From the Python docstring for rebuild():

```
        Rebuild the shareref table.
        Clears the shareref table and then rebuilds it by iterating over all Ganga Objects
        in the Job, Box and Task repositories. If an object has a ShareDir associated with it,
        that ShareDir is added into the shareref table (the reference counter being incremented
        accordingly). If called with the optional parameter 'unprepare=False', objects whose
        ShareDirs are not present on disk will not be unprepared. Note that the default behaviour
        is unprepare=True, i.e. the job/application would be unprepared.
        After all Job/Box/Task objects have been checked, the inverse operation is performed,
        i.e., for each directory in the ShareDir repository, a check is done to ensure there
        is a matching entry in the shareref table. If not, and the optional parameter
        'rmdir=True' is set, then the (orphaned) ShareDir will removed from the filesystem.
        Otherwise, it will be added to the shareref table with a reference count of zero;
        this results in the directory being deleted upon Ganga exit.
```

Additionally, a lookup can be performed for a given ShareDir directory to discover which jobs/applications/tasks refer to that directory. Optionally, the resulting objects can be unprepared too. Again, from the GPI, calling:

```
In [12]:shareref.lookup('conf-70861947-6038-1333531939-34')
Ganga.GPIDev.Lib.Registry          : INFO     ShareDir conf-70861947-6038-1333531939-34 is refere
Ganga.GPIDev.Lib.Registry          : INFO     ShareDir conf-70861947-6038-1333531939-34 is refere
Ganga.GPIDev.Lib.Registry          : INFO     2 item(s) found referencing ShareDir conf-70861947-
```

or

```
In [13]:shareref.lookup('conf-70861947-6038-1333531939-34', unprepare=True)
Ganga.GPIDev.Lib.Registry          : INFO     ShareDir conf-70861947-6038-1333531939-34 is refere
Ganga.GPIDev.Lib.Registry          : INFO     Unpreparing job repository object #1015 associated
Ganga.GPIDev.Lib.Registry          : INFO     ShareDir conf-70861947-6038-1333531939-34 is refere
Ganga.GPIDev.Lib.Registry          : INFO     Unpreparing box repository object #204 associated w
```

It is possible to unprepare every prepared job/application/task using the following loop:

```
for conf in shareref._impl.name.keys():
    shareref.lookup(conf, unprepare=True)
```

This topic: ArdaGrid > GangaPreparedApplications
Topic revision: r39 - 2012-04-04 - MichaelJohnKenyonExCern