

# Cobalt 0.98 Release Notes

This page details the current release status and upgrade instructions for Cobalt-0.98.

## Release Status

cobalt-0.98.1 has been released on 4/03/2008. This release has been stress tested on the 40 rack BG/P system at ANL, in addition to some smaller systems here as well.

Binary rpms for sles9/ppc64 are available from the [Cobalt FTP site](#).

Binary rpms for sles10/ppc64 are available from the [Cobalt FTP site](#).

## Known Issues

- No known issues exist at this time

## Differences from previous versions

- Cobalt has started using the bridge api directly. This means that the DB2 python extension is no longer required
- Use of the bridge api requires both a 64-bit python and ctypes. We have built 64-bit python RPMS for both BG/L systems (SLES9) and BG/P systems (SLES10).
- tsslite is required
- Cobalt now determines all partition information directly from the control system. This means:
  - ◆ partition sizes don't need to be manually specified
  - ◆ partition dependencies don't need to be manually specified
  - ◆ partadm now displays reasons for partition blockages
- Queues have a Priority attribute

## Running 0.98.1

If you are currently running Cobalt 0.97, we have provided some scripts that can help you recreate the state of your system when upgrading to Cobalt 0.98.1. Before installing the Cobalt 0.98.1 rpms, you will need to download a few client scripts that will talk to your 0.97 components and generate commands that can be issued to your new Cobalt 0.98.1 install.

Use something like wget or curl to fetch the following scripts:

[https://svn.mcs.anl.gov/repos/cobalt/tags/cobalt\\_0\\_98\\_1/misc/mk\\_jobs.py](https://svn.mcs.anl.gov/repos/cobalt/tags/cobalt_0_98_1/misc/mk_jobs.py)

[https://svn.mcs.anl.gov/repos/cobalt/tags/cobalt\\_0\\_98\\_1/misc/mk\\_partitions.py](https://svn.mcs.anl.gov/repos/cobalt/tags/cobalt_0_98_1/misc/mk_partitions.py)

[https://svn.mcs.anl.gov/repos/cobalt/tags/cobalt\\_0\\_98\\_1/misc/mk\\_queues.py](https://svn.mcs.anl.gov/repos/cobalt/tags/cobalt_0_98_1/misc/mk_queues.py)

Each script will contact your Cobalt 0.97 installation and retrieve information about jobs, partitions, and queues, respectively. The scripts gather information from Cobalt and then print commands to standard out that should recreate the current state of the system. Save the output from each command for use after installing the Cobalt 0.98.1 rpms.

If you are upgrading from Cobalt 0.98.0, your existing state files should work just fine.

At this point, you can install the Cobalt 0.98.1 rpms. The `bgsystem` component should start quickly on a one rack system, but larger systems may exhibit a slight delay. At startup, `bgsystem` is making a call to the bridge API to find out information about the machine as a whole, and this obviously takes longer the more machine there is.

Once all of the components are running, you should be able to execute the commands produced by the upgrade scripts (or enter your own commands if you aren't upgrading).

First, tell Cobalt which partitions it should manage. Note that, through the bridge, Cobalt is aware of all partitions on the system, but it will only attempt to run jobs on the partitions that you have told it to manage through `partadm`. `partadm` only needs the `-a` flag now, as the size and dependencies are automatically discovered through calls to the bridge API.

Next, create the queues. This works the same as in Cobalt 0.97, although note that queues now have a Priority attribute. Jobs in queues with higher priority will be given the opportunity to execute before jobs in queues with lower priority.

Finally, if you are migrating from Cobalt 0.97, run the commands that will recreate the jobs in your queues.

You will also need to recreate your reservations. Note that the reservation system has changed in Cobalt 0.98.0. The `-a` and `-x` flags for `setres` are gone. When you create a reservation, think of it as a reservation of node cards. Jobs in the reservation queue can use any subpartition of that reservation's partition. No outside job will be allowed to start that would block any of the reservation's partitions or subpartitions. `setres` also has a new `-q` flag if you wish to specify a particular queue for the reservation, as well as a `-c` flag to specify a cycle time for automatically repeating reservations.