# OASIS: OSG Application Software Installation Service

## Introduction

This document outlines the motivation and requirements for the OSG Application Software Installation Service (OASIS). OASIS is a value-added service the OSG may offer to smaller VOs to assist with the distribution of VO application software. This service would not be mandatory, but we believe would significantly lower the barrier in becoming an OSG VO.

The basic concept is the OSG will offer a central login to a software manager for each VO. The manager will be able to login and install their application software in a central location. Once installed in the central location, the OSG is responsible for distributing the VO software to as many sites as possible. In exchange for using the OSG-provided service, the VO will have to abide by a certain set of restrictions such as total space used and number of files.

The requirements outlined in this document cover what we believe is needed for the first version of OASIS; we highlight requirements that may be done in the future (implicitly, any such requirements are not part of the first version).

## Background

The OSG has always encouraged the concept of software portability: a user’s scientific application should be able to run in as many operating system environments as possible. This is typically accomplished by compiling the software into a single static binary, or distributing all the dependencies in a tarball downloaded by each job. However, the concept of portability runs against the current Linux software distribution philosophy, and becomes increasingly difficult to achieve as the size of a scientist’s software stack increases. At this point, “portability” is a barrier adoption of the OSG and it is not pragmatic to provide software packaging assistance to every OSG VO.

Accordingly, it is necessary to provide a mechanism for OSG VOs to pre-install software at the sites where they run.

Since its first release, the OSG CE has provided a directory to VOs, referred to as “OSG APP” (for the job environment variable, $OSG\_APP, that points to its runtime location), that can be used for application software installations. The VO assumes it can create a sub-directory that it “owns”, install its software into the sub-directory, and have the directory shared on the worker nodes for a site. The OSG provides guidelines for the size and Unix permissions of the $OSG\_APP directory.

For example, CMS installs its applications inside $OSG\_APP/cmssoft. The initial installation takes about 10GB of disk space, and each subsequent install can take another 1GB. Only CMS has the ability to write into $OSG\_APP/cmssoft.

The $OSG\_APP model has shown to have a few issues:

* It provides no enforcement or allocation mechanism. A single “badly-behaved” VO can utilize all the space available, denying service to other “good” VOs. When this happens, the local site has to make decisions about how to clean up the shared space.
  + The site likely has an internal prioritization for one or two VOs. An ATLAS site will give ATLAS priority in $OSG\_APP, but likely does not care about the rest of the OSG VOs.
* It may be implemented inconsistently across sites. Some sites prefer VOs do the installation from the worker nodes; others have $OSG\_APP read-only from the worker nodes. VOs typically have to learn the idiosyncrasies site-by-site, and there is little sharing between VOs.
* OSG provides no toolset for distributing software. Sites provide a writable directory, but VOs want higher level concepts such as “copy this file to all accessible sites”.

## Requirements for OASIS v1

1. **Space allocation**. VOs using this service must abide by some space allocation restrictions. An example may be the total space used by the software install, but . These policies must be enforced by the login service, and will be set by the OSG ET.
2. **Login node**. OASIS will provide a central node with SSH access for the VOs. This node will have sufficient space for the VO to utilize its full allocation. Each VO will be able to install its software into a given directory and indicate to OASIS when their directory is ready to be synchronized.
3. **Authorization managemen**t. The OASIS Operations team must provide a login node and an authorization scheme such that the VO can restrict SSH access to the node.
4. **Open Software**. VOs may only use OASIS for software that can be freely redistributed by the OSG Operations team to an unlimited number of sites. For the initial version, software installed via OASIS will be accessible to other VOs.
5. **Periodic synchronization**. OASIS will attempt to periodically synchronize resources with the central repository. This synchronization is “best-effort”; OASIS will attempt to keep all resources synchronized within a certain time window, but can provide no guarantees. Integrity – the size, permissions, and contents of the directories – will also be best-effort. Integrity cannot be guaranteed for any point in time on the worker node, but will be checked after synchronization.
6. **Availability**. The VO software will be available on the worker node at a deterministically-determined directory location. This location might be relative to an environment variable ($OSG\_APP/oasis/cms), or might be an absolute path (/oasis/cms).
7. **Atomicity**. The software present on the worker should be consistent with a single version of the directory on the central repository. Partial updates should not be visible to user jobs.
8. **Registry/Versioning**. OASIS should keep a registry of the resources where the software install was attempted, and the state of the synchronization. This state should be available from the worker node without having to query a centralized service.

## Possible Future Requirements

1. **Delegation**. VOs can delegate their credentials to OASIS, so the software can be installed on the sites using the VO’s authorization, instead of the OASIS credentials.
2. **Closed Software**. If a VO delegates their credentials to OASIS, it would be possible to distribute software only readable by that VO. Beyond the technical issues, there are a myriad of licensing issues that may still make this impossible to provide.
3. **Customization**. Many pieces of software require customization if they are copied into a directory different from the original install directory. A future version of OASIS may allow the VO to provide a customization script that is run after the software is synchronized.
   1. For example, Pacman installs provide an environment setup file that references the absolute path of the original install location. If $OSG\_APP is not the same absolute directory on all sites, pacman-based installs will not work by just copying to the remote site – they must be customized.
   2. To do this securely, it is likely is dependent on delegating the VO credentials to OASIS.
4. **Integrity**. The first version of OASIS provides best-effort verification of the software install. A future version may provide integrity, guaranteeing that the client only reads the correct version of the installed file.

## Technologies

This section is to be filled in when requirements are finalized. Until then, we try to record a few important “decision points” that will need to be made.

**Push versus Pull**: Should the software be “pushed” to the site by OASIS, or “pulled” upon demand when the first job runs? If it is pulled, what caching hierarchy must be in place? What cache management and thrashing needs to be monitored?

**Site requirements**: What site support is required for OASIS? If there is off-the-shelf technology that requires site-level changes, what priority should be given to that versus developing a new software project? Is it acceptable to have OASIS depend on new OSG software versions, versus having it available at all sites?