

# IceGrid Persistent Data

The IceGrid registry and node both store information in the data directories specified by the [IceGrid.Registry.Data](#) and [IceGrid.Node.Data](#) properties, respectively. This section describes what the registry and node are storing and discusses backup and recovery techniques.

On this page:

- [Registry Persistent Data](#)
- [Node Persistent Data](#)

## Registry Persistent Data

The registry's data directory is a [Freeze](#) database environment that stores the following information:

- Applications [deployed](#) using the `addApplication` operation on the `IceGrid::Admin` interface (which includes the IceGrid GUI and command-line [administrative clients](#)). Applications specify servers, well-known objects, object adapters, replica groups, and allocatable objects. Applications can be removed with the `removeApplication` operation.
- [Well-known objects](#) registered using the `addObject` and `addObjectWithType` operations on the `IceGrid::Admin` interface. Well-known objects added by these operations can be removed using the `removeObject` operation.
- [Adapter endpoints](#) registered dynamically by servers using the `Ice::LocatorRegistry` interface. The property [IceGrid.Registry.DynamicRegistration](#) must be set to a value larger than zero to allow the dynamic registration of object adapters. These adapters can be removed using the `removeAdapter` operation.
- Some internal proxies used by the registry to contact nodes and other registry replicas during startup. The proxies enable the registry to notify these entities about the registry's availability.

[Client session](#) and [administrative sessions](#) established with the IceGrid registry are not persistent. If the registry is restarted, these sessions must be recreated. For client sessions in particular, this implies that objects allocated using the allocation mechanism will no longer be allocated once the IceGrid registry restarts.

If the registry's database is corrupted or lost, you must recover the deployed applications, the well-known objects, and the adapter endpoints. You do not need to worry about the internal proxies stored by the registry, as the nodes and registry replicas will eventually contact the registry again.

Depending on your deployed applications and your use of the registry, you should consider backing up the registry's database, especially if you cannot easily recover the persistent information.

For example, if you rely on dynamically-registered adapters, or on well-known objects registered programmatically via the `IceGrid::Admin` interface, you should back up the registry database because recovering this information may be difficult. On the other hand, if you only deploy a few applications from XML files, you can easily recover the applications by redeploying their XML files, and therefore backing up the database may be unnecessary.

Be aware that restarting the registry with an empty database may cause the server information stored by the nodes to be deleted. This can be an issue if the deployed servers have databases stored in the node data directory. The next section provides more information on this subject.

## Node Persistent Data

The IceGrid node stores information for servers deployed by IceGrid applications. This information is stored in the `servers` subdirectory of the node's data directory. There is one subdirectory per server; the name of the subdirectory is the server's ID. Each server directory contains configuration files, and may also contain [database environments](#) and the [distribution data](#) of the server. The node's data directory also contains a `distrib` directory to store per-application distribution data. This directory contains a subdirectory for each application that specifies a distribution and has a server deployed on the node.

If a server directory is deleted, the node recreates it at startup. The node will also recreate the server configuration files and the database environment directories. However, the node cannot restore the prior contents of a server's database environment. It is your responsibility to back up these database environments and restore them when necessary. If the server or application distribution data is deleted from the node's data directory, you can easily recover the deleted information by patching these distributions again using the IceGrid administrative tools.

If you store your server database environments outside the node's data directory (such as in a directory that is regularly backed up), or if you do not have any database environments inside the node's data directory, you do not need to back up the contents of the node's data directory.

See Also

- [Freeze](#)
- [Using IceGrid Deployment](#)
- [icegridadmin Command Line Tool](#)
- [Well-Known Objects](#)
- [Object Adapter Endpoints](#)

- [Resource Allocation using IceGrid Sessions](#)
- [IceGrid Administrative Sessions](#)
- [DbEnv Descriptor Element](#)
- [Application Distribution](#)
- [IceGrid Properties](#)